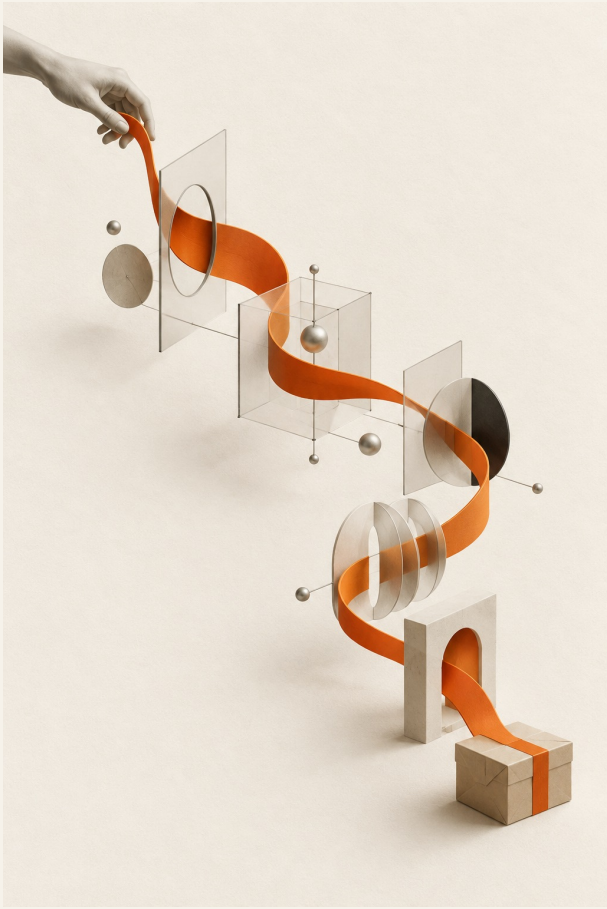
STEFAN HAMANN

STRATEGY / TECHNOLOGY / LEADERSHIP

AGENTIC COMMERCE

The Handbook for Commerce Strategy, Technology and Leadership



2026 · ENGLISH EDITION

Supported by

shopware

ACA Agentic Commerce Alliance

Agentic Commerce

Stefan Hamann

The Handbook for Commerce Strategy, Technology and Leadership

Co-founder of Shopware · Initiator of the Agentic Commerce Alliance

English edition · Revised and expanded
29 October 2026 · Editorial revision R10

This manuscript connects thematically relevant publications with current primary sources. Diagrams are original visualizations; dated facts and model calculations identify their sources or assumptions.

Self-published by Stefan Hamann, Kronenweg 16, 48683 Ahaus, Germany.
First edition 2026.
Trademarks and quoted works remain with their respective rights holders.

agentic-commerce.sh

About the author



Stefan Hamann is co-founder and Chief Agentic Officer of Shopware. For more than 25 years, he has shaped the company's technological development. Today, he and his team lead research and practical implementation in agentic commerce. [S145]

Stefan Hamann on stage. Author photograph: agentic-commerce.sh.

He published his first articles and a whitepaper on agentic commerce in January 2025. His work combines a clear position with practical experiments: merchants should be able to use new technology to develop businesses of their own and strengthen their customer relationships.

As the initiator of the Agentic Commerce Alliance, he works to give merchants a voice in this shift. At Shopware's Agentic Commerce Lab, he connects this perspective with product development. This book brings his theses, experience and experiments together in a handbook for the next phase of commerce.

Contents

The shift and its economic logic **7**

01 When the customer orders an outcome 8

02 Why agentic commerce matters now 13

03 Seven theses for commerce with agents 20

Customers, brands and experience **25**

04 The new customer journey 26

05 Being found before anyone visits the shop 30

06 Experience that improves a decision 36

The purchasing journey and its protocols **47**

07 How protocols become a purchasing journey 48

08 From the offer to the service that follows 53

09 When purchasing moves between interfaces 64

Operations, responsibility and economics **68**

10 Architecture and B2B in daily operations 69

11 From copilot to autonomous shop operations 76

12 Control and law 88

13 Security for agents that act 93

14 The economics behind the agent 97

Build it yourself **102**

15 Open code becomes working material for agents 103

16 When merchants build for themselves 111

17 How we learn in the Agentic Commerce Lab 117

Act strategically **123**

18 From insight to action with the Alliance 124

19 The first 90 days and the choice of partners 130

20 A reason to buy from you 136

21 Four futures of commerce 146

Reference and working materials **152**

A Terms from A to Z 153

B The compact protocol passport 162

C Five working templates for leaders 165

D Eight practical implementation patterns 167

E How the published theses developed 169

F Sources and further reading 175

Foreword

For more than 25 years, I have been building software for merchants at Shopware. One tension keeps returning: shared infrastructure needs to be economical, while each business needs room to make something of its own. The most interesting ideas often begin where the standard ends. For many merchants, implementing them was too expensive.

Agents change that calculation. They take shopping chores off customers' hands and help merchants develop their own software, knowledge and operations. What interests me is the entrepreneurial freedom this creates. Merchants should retain control of their businesses and build offers that people actively choose. I have been writing about this since January 2025, when agentic commerce was still unfamiliar to much of the industry.

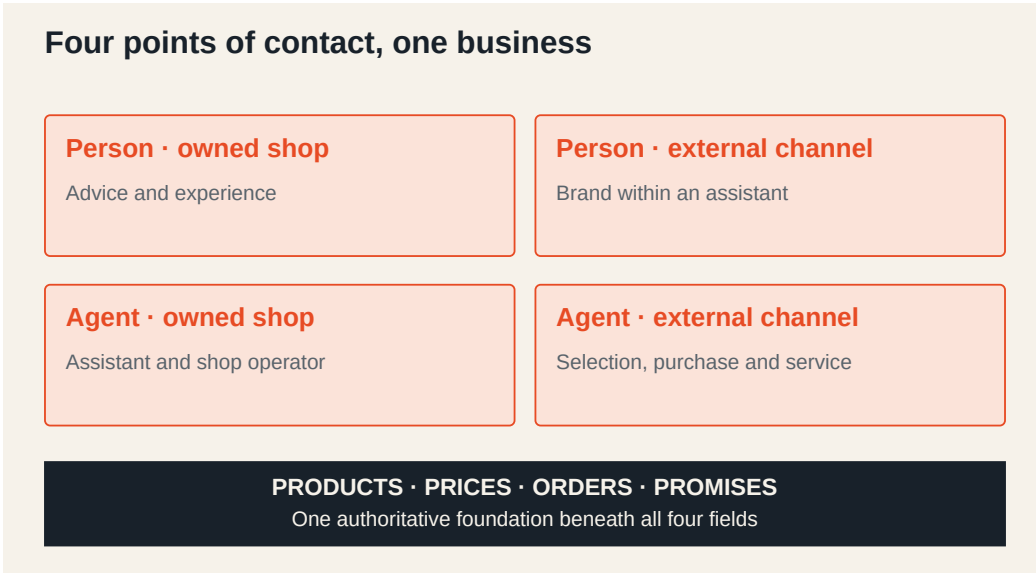
At Shopware, we are developing the platform towards commerce in which agents take on more of the work. Our Agentic Commerce Lab explores new purchasing journeys and tools. Separately, I use VEYNDO as my private experimental platform for new concepts, from product knowledge to autonomous merchant operations. The book presents these projects in their respective contexts. Their repositories are listed in the appendices. Readers who disagree will often find somewhere to start a pull request.

We begin with a suitcase that needs to arrive by Friday. That simple request connects product data, selection, price, consent, payment and delivery. The traveller and her suitcase accompany us through the book, all the way to the airport on Friday. Along the way, we also explore technical spare parts, room design and how merchants can turn new possibilities into a good business.

The development of my thinking, notes on the examples and a guide to the public experiments are collected in Appendix E.

The shift and its economic logic

A customer request connects selection, approval and delivery. Agents change who does that work and who controls customer access. The opening chapters develop the economic logic and seven theses for commerce.



CHAPTER 01

When the customer orders an outcome



The request is clear. The suitcase is already packed.

From clicking to a request

The suitcase has to arrive by Friday. It must cost no more than €180, meet the cabin-baggage rules of the airlines on the trip and roll quietly. The traveller gives her assistant the task. She wants to get on with planning her trip.

For the merchant, that request brings together the right model, a complete price and a delivery promise it can keep. The customer delegates the search; the business needs to describe and connect its services so an agent can use them. We will follow that division of work all the way to the airport.

An agent can research, compare alternatives and prepare an order; its mandate determines whether it may complete the purchase. That distinction turns a compelling demo into a responsible business process: the objective describes the desired result, while the mandate limits permitted actions. The system must record them separately.

Where value moves

Competition begins before the visit to the shop. Whoever understands the request and assembles the shortlist influences which merchants are considered. Advice may happen inside an assistant while the merchant manages the offer, contract, stock, shipping and complaints. For the merchant, connecting this new customer interface to the business becomes essential.

My early writing about AI in commerce emphasized relief from operational complexity. Later work focused more strongly on the distribution of decision power. These perspectives belong together. A merchant can use agents to become more productive and still lose market share because external agent systems cannot access its offers effectively. Internal efficiency and external accessibility are separate investment fields. [S01, S02, S06, S15]

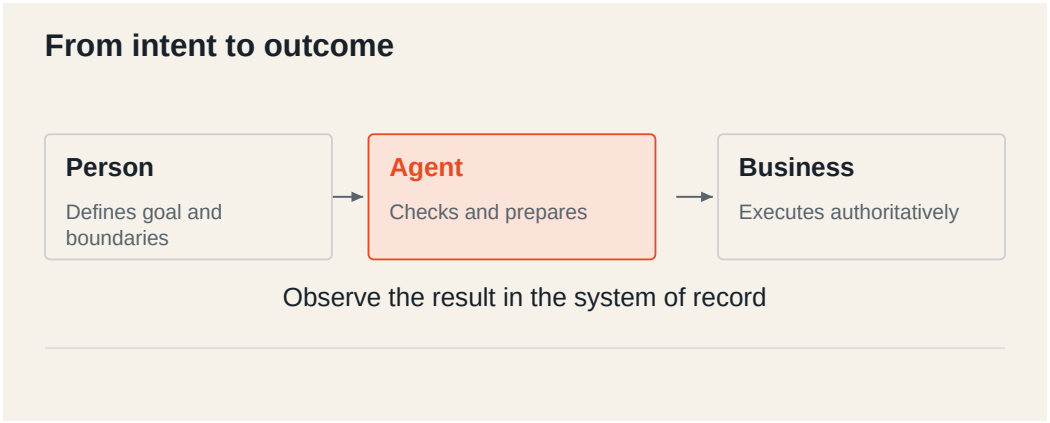


Figure 1. From customer request to delivery: person, agent and business contribute to the outcome.

Agents change both sides of competition. Customers can compare offers more easily. Merchants can use coding agents to build more of their own ideas. Bringing the two together means investing in a reason to buy and in the ability to deliver it reliably.

The traveller is buying a suitcase and the prospect of reaching the airport on Friday with the right luggage. A workshop orders a spare part so its machine can run again the next morning. Seen this way, commerce includes both the product and the service that makes it valuable to the customer.

Departments have traditionally worked on different parts of a purchase: product teams on suitability, purchasing and logistics on availability, and the shop on ordering. An agent needs those contributions together because it must establish what fits, what is available and what it costs. The merchants that win will bring their best knowledge and most dependable commitments together in the same task.

One suitcase, six transitions

The traveller in the opening needs her suitcase by Friday. Follow the request through to delivery. Each transition asks something different of the system; these transitions recur throughout the book.

Step	Who acts	What is needed	Typical break
Request	Traveller and personal agent	€180, Friday, dimensions, quiet wheels	No evidence for “quiet”
Selection	Agent	Product dimensions and current airline rules	External dimensions missing
Offer	Merchant	Total price and delivery commitment	Shipping exceeds the budget
Consent	Traveller	Specific offer revision	Price changes afterwards
Order and payment	Shop and payment service	Unique request and payment status	Timeout triggers a duplicate order
Fulfilment	Merchant and logistics	Delivery, exceptions and service	Delay never reaches the agent

The language model understands what the traveller wants. Product data, delivery commitments and a reliably stored order determine whether the suitcase arrives by Friday. We will revisit the same questions at Nordstern in outdoor shopping, Werkpunkt in spare-parts procurement and Lichtfeld in room design.

Agentic commerce begins when software takes on a request and organises steps to fulfil it. A customer describes the desired result. An agent gathers information, compares options, asks questions where needed and carries out the actions it is allowed to take. This can happen while shopping, in daily shop operations or between companies.

It needs a clear request, useful information and tools for acting. It also needs a boundary: what can it decide itself, and when should it ask a person? A good assistant asks when it reaches that boundary and completes the other steps itself.

Assign permissions to each action

An agent may read information, prepare a basket, order after customer consent or replenish known supplies under a mandate. Assign these permissions to each action. The same agent can have broad reading access and narrow purchasing authority.

Action	Permission	Boundary
Read	Retrieve products and delivery options	Approved data only
Prepare	Draft a basket or change	No order or change submitted yet
Execute with consent	Order a specific offer	Consent to this revision
Act under a mandate	Replenish supplies	Amount, products, period and exceptions

The operating model in Chapter 11 concerns how a whole shop develops from assistance towards work directed by goals. Permission for one purchase and autonomous shop operations are different decisions.

What remains useful about existing automation

Rule-based automation remains valuable. A fixed approval step does not need to become a language-model task. Agentic systems are especially useful when inputs vary, information is distributed or the next sensible action is not fully predetermined. A conventional integration process is often simpler and more reliable for a stable, repetitive data transfer.

A strong system combines approaches. The agent interprets “We need the right accessories for the new workstations again.” Search identifies compatible items. Pricing calculates customer-specific terms. Rules check budgets and delivery addresses. A person approves the first order. No component needs to pretend it can do everybody else’s job.

A **buyer agent** represents its principal’s needs and constraints. A **merchant agent** represents the seller’s products, services and permitted offers. An **operations agent** supports internal work such as catalogue maintenance or service preparation. Their interests are not identical. A buyer agent should not quietly become a sales representative because a merchant supplies particularly attractive context.

Intermediaries can aggregate demand, select offers, facilitate payments or establish trust. Because remuneration affects incentives, an agent that both recommends a purchase and earns from its completion needs transparent rules for that dual role.

From a promise to a completed purchase

In a demonstration, the traveller should be able to find and order her suitcase. Then one condition changes: the preferred model sells out or delivery takes an extra day. The assistant has to revise the selection, seek consent or bring in customer service. That is where we see whether it can work within the merchant's business.

The same task can arise in four places. The traveller may browse the shop herself or consult an assistant within it. Outside the shop, she may ask her personal agent or discover the product in another interface and decide for herself. For the merchant, these routes belong to one business and need a connecting foundation.

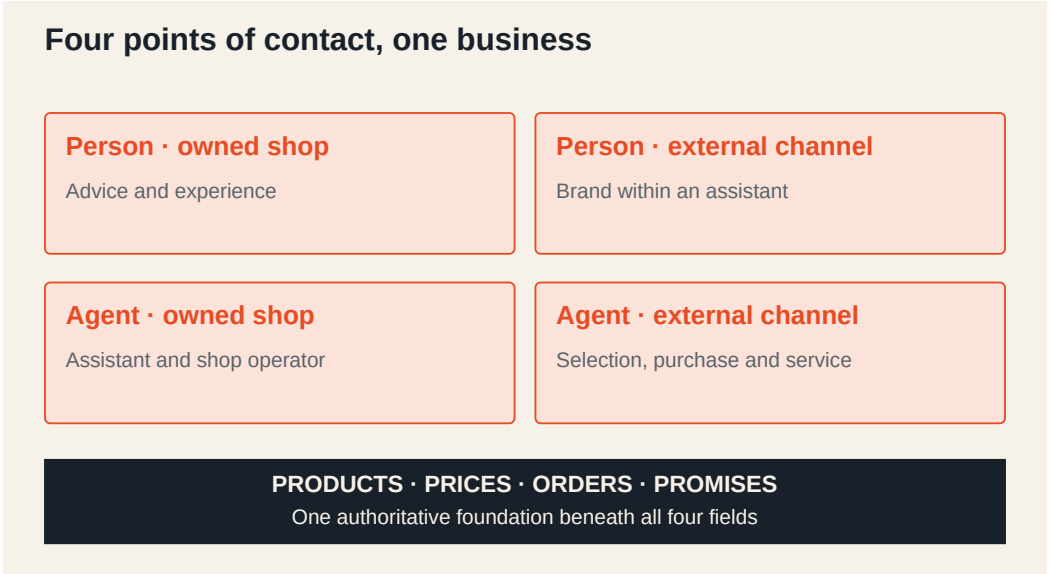


Figure 2. People and agents use owned and external channels. Products, prices, orders and promises connect all four points of contact.

This map keeps strategy from becoming too narrow. Serving only external agents leaves advice and experience in the merchant's own shop unattended. Adding only a chatbot misses customers in other assistants. All four fields need the same product records, prices and orders. How quickly demand moves between them shapes the next investment.

CHAPTER 02

Why agentic commerce matters now



The infrastructure is taking off. The suitcase is not waiting for another forecast.

The infrastructure has become concrete

A merchant can now give customers another way into its business. They search with an assistant, have an offer prepared and move to the shop to buy, or stay in an application that supports the purchase. Shared connections to product data, tools and commerce functions make this possible. MCP, ACP and Instant Checkout, introduced in 2025, are part of that infrastructure; Google added UCP in January 2026. [S30, S25, S27]

Major distribution and payment providers are building these connections too. Microsoft presented Copilot Checkout and other retail capabilities in January 2026. PayPal’s Agent Ready and Store Sync connect product access and payment, while Stripe develops the payment side further. Merchants gain practical routes from existing systems to new purchasing interfaces. [S64, S47, S42]

Selection can begin in an assistant and become a purchase in the shop. OpenAI's change of direction on 24 March 2026 illustrates this division of work. The company said its first Instant Checkout implementation had offered too little flexibility and placed greater emphasis on product discovery alongside merchants' own checkout experiences. [S26]

The strategic opportunity lies in a purchase distributed across several places. A customer discovers an offer in one, resolves a question in another and completes the purchase where the appropriate functions are available. Merchants that design these transitions well can use existing commerce infrastructure in new customer channels. [S07]

Standards are also emerging to demonstrate that a person consented to a particular purchase. In May 2026, AP2 and Verifiable Intent entered the FIDO Alliance's standardisation work. Mastercard continues to develop agent recognition and integration. Personal agent protocols, including the approach announced by Meta and Sierra in October 2026, address the longer relationship between customer and company. [S39, S38, S41, S33, S34]

Together, these developments connect the journey: find offers, establish terms, obtain permission, pay and resume a service request later. Chapters 7 to 9 explain each approach through its part in that journey. The protocol passport records the dated development snapshots. [S36]

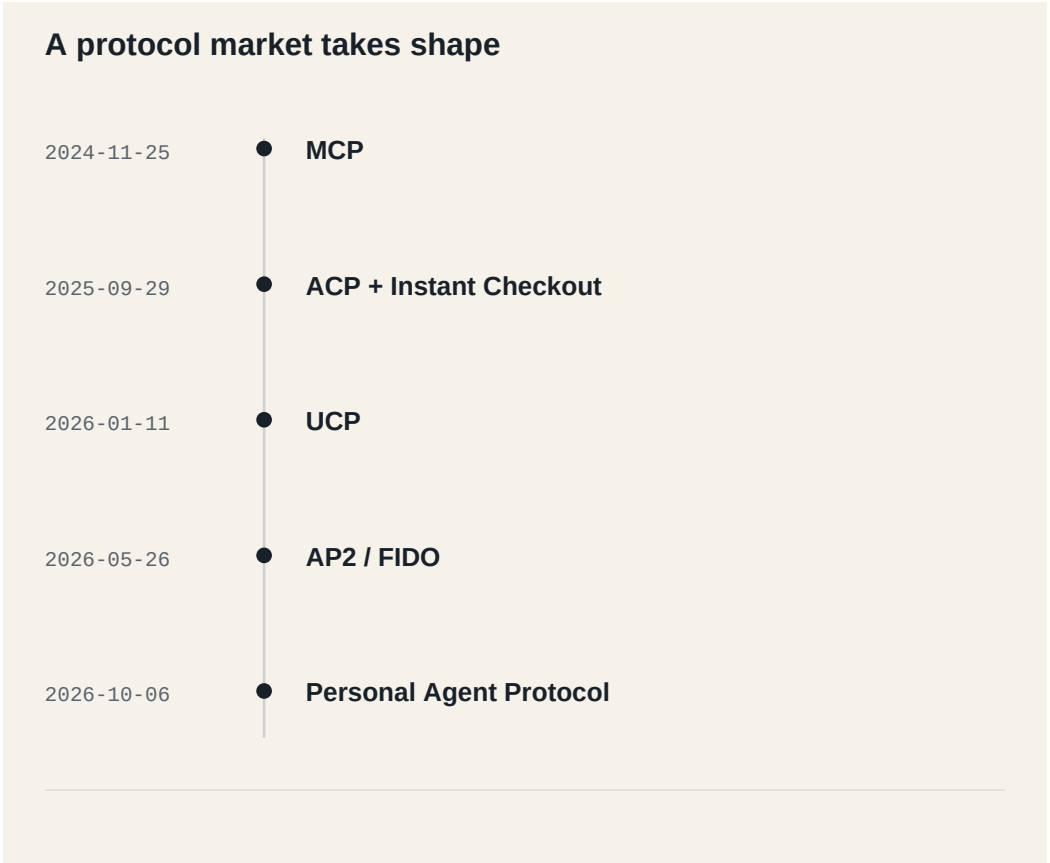


Figure 3. From the MCP launch to personal agents: selected announcements and standardization milestones. Sources S25, S27, S33, S39, S143.

Why this matters beyond another interface

Agentic commerce can change where demand forms, what intermediation costs and who owns the customer relationship. Market forecasts show the scale that strategy firms and analysts see in that change.

In October 2025, McKinsey estimated that agentic commerce could orchestrate USD 3 trillion to USD 5 trillion in global consumer-goods revenue by 2030. Its US B2C estimate was USD 900 billion to USD 1 trillion. The scenario concerns consumer-goods revenue organised through agents; services and B2B sit outside it. [S79]

Bain expects agents to account for 15–25 percent of US e-commerce by 2030, equivalent to USD 300–500 billion. Unlike some broader estimates, its scenario covers purchases initiated, influenced or completed by third-party or merchant agents

and excludes AI search and discovery alone. That definition explains part of the range. [S80]

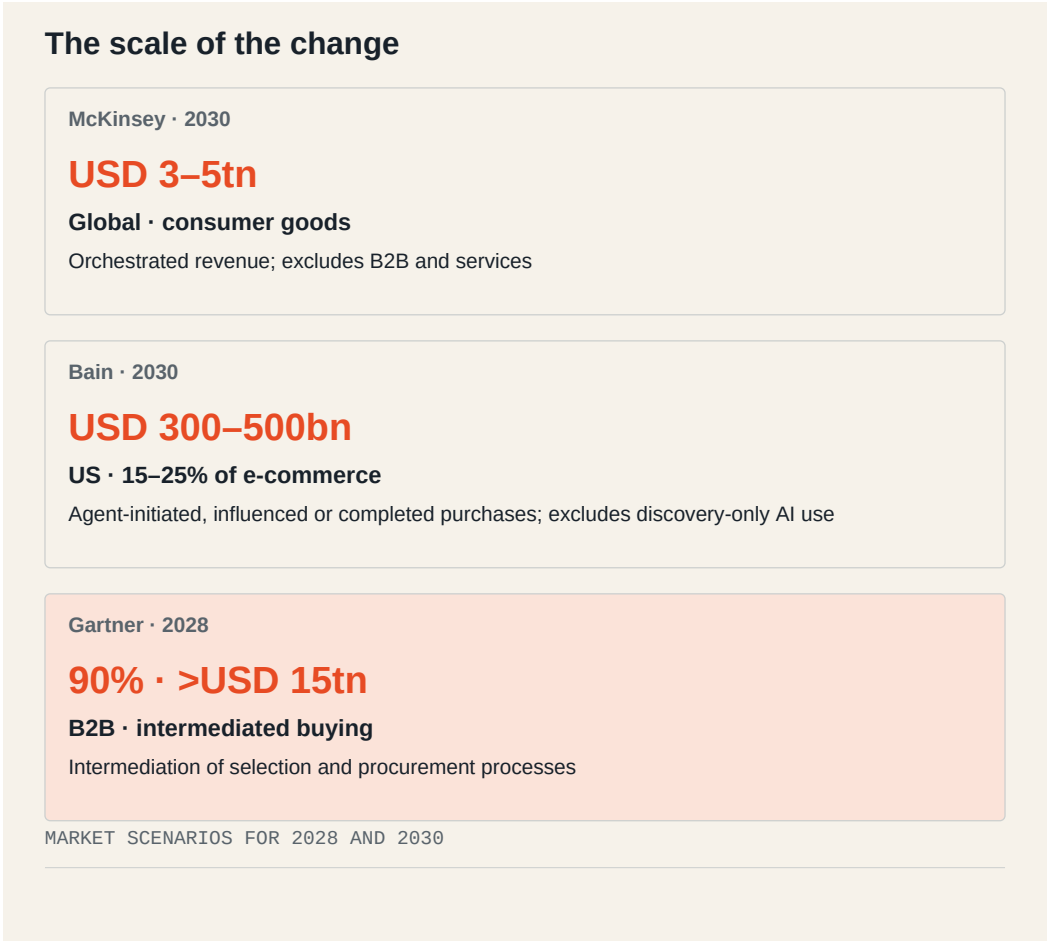


Figure 4. Market forecasts for consumer goods, US e-commerce and B2B procurement. Sources S79, S80 and S81.

Gartner presents a particularly far-reaching B2B scenario: by 2028, agents could intermediate 90 percent of purchasing, with more than USD 15 trillion passing through agent exchanges. Here, mediation includes preparing and processing procurement. The forecast does not specify how often a person still approves the individual purchase. [S81]

Gartner also expects 60 percent of brands to use agentic AI for personal interactions in marketing, sales and service by 2028. At the same time, it predicts that more than 40 percent of agentic-AI projects will be cancelled by the end of 2027 because of cost, unclear value or inadequate risk controls. Both predictions lead to the same

leadership question: where does the technology improve the business enough to sustain its use? [S82, S83]

These figures describe scenarios for different markets and activities. Different as the figures are, they point in the same direction: who does the purchasing work is becoming an investment question. For an individual merchant, the answer begins with its customers, products and economics.

What is already being observed

Early changes are already visible in shopping behaviour. In a McKinsey survey of nearly 1,000 European consumers published in March 2026, 38 percent said they used AI to discover products and brands or support purchase decisions. Change begins before the purchase itself. [S84]

Adobe reports a 1,439 percent increase in referrals from AI platforms to UK retail sites between October 2024 and August 2026. In August, the conversion rate for these visitors was around 20 percent higher than for non-AI-referred traffic. AI assistants are sending measurable traffic to UK shops, and those visitors convert more often. The figure does not show whether AI causes the difference or whether these visitors were already more determined to buy. [S85]

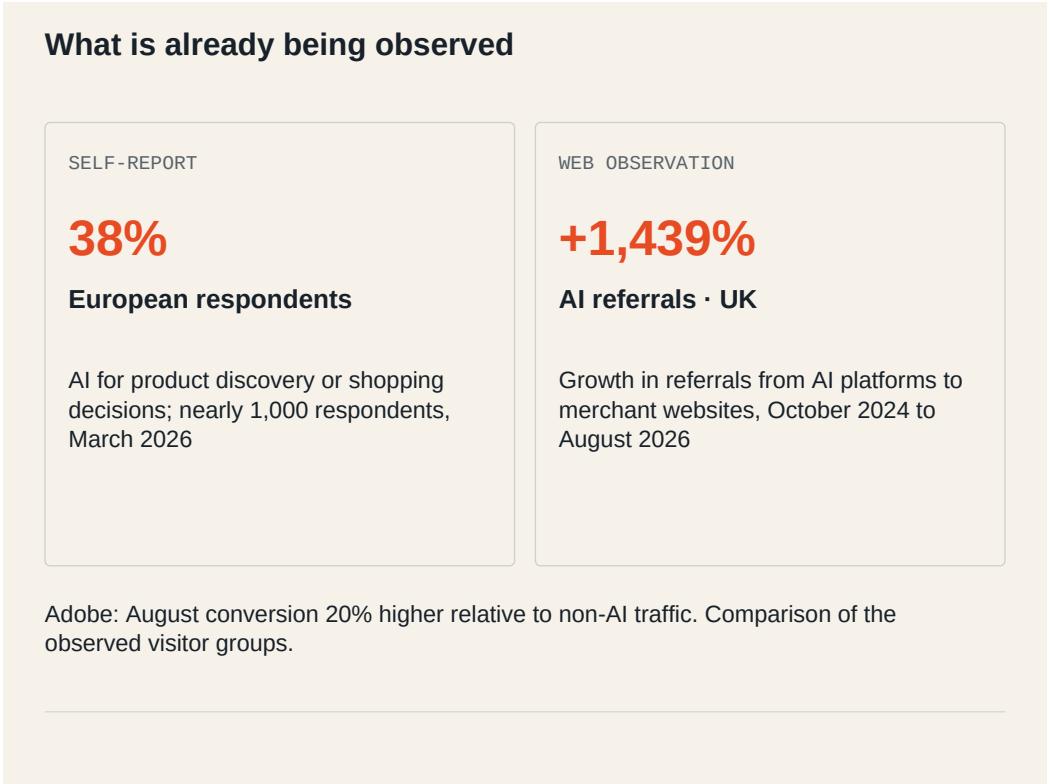


Figure 5. AI already influences product selection and referrals. Findings from a European survey and UK web observations. Sources S84 and S85.

A merchant can follow the change in its own range. Marketing tracks recommendations and enquiries from assistants. Accounting connects the resulting orders with fees and service costs. Together, those records show how much the new route contributes to the business.

Investing as the market develops

My bet is that shortlisting will concentrate among a few assistants before purchasing becomes fully autonomous. I would therefore invest in discoverable offers and dependable purchasing journeys today; Chapter 21 develops that position.

For a spare-parts merchant, a few additional enquiries prepared correctly each week may justify an integration. A fashion merchant might invest in selection support first because customers still place orders themselves. The customer’s task determines where the money goes.

Some investments pay off regardless of market speed: dimensions and delivery commitments help the next customer, whether a person or an agent. A second entry

point becomes valuable when customers use it, while an exclusive intermediation agreement ties the business to one operator. Begin with improvements that hold across scenarios. The following theses explain the forces shaping that decision.

Where to begin in your business

The first change depends on the customer's task: reordering familiar items, finding a compatible spare part or discovering something new. The following overview translates those differences into initial investments.

The first investment depends more on the customer's problem than on the next protocol announcement. For five common business models, I would start here. These are strategic recommendations drawn from the book's arguments.

Merchant type	First change	First investment	Greatest risk	Read on
B2B, spare parts, industry	Purchasing agents request compatibility and customer prices	Verified compatibility and protected price access	Wrong fit; intermediaries aggregate demand	Chapters 10, 16
Fashion and lifestyle	Selection in the assistant, often followed by a personal purchase	Fit data, return reasons and designed experience	Interchangeable branded goods and avoidable returns	Chapters 5, 6, 20
Interiors and furniture	Inspiration and planning with an assistant	Dimensions, room visualization and handover to advisers	A design promises more than the product delivers	Chapter 6
Replenishment and consumables	Familiar purchases are delegated	Product identity, mandates and repeat ordering	Price comparison displaces the established relationship	Chapters 1, 8, 14
Brand with its own shop	Customers name the brand in their instructions	Verifiable promises and designed experience across interfaces	Other sellers offer the same goods in the same assistant	Chapters 3, 6, 20

Many businesses occupy several rows. Choose the customer task whose improvement can first make a difference: the right spare part, greater confidence about fit or effortless replenishment.

CHAPTER 03

Seven theses for commerce with agents



Seven theses. One rather ambitious handle.

Two changes, one strategic task

The agent takes work off the customer’s hands. It gives the merchant a new challenge: the offer must be convincing before the customer has even seen the shop. At the same time, coding agents help build distinctive services that used to cost too much. This connection runs through my theses: comparison gets easier, building gets easier, and a merchant needs a clearer reason for customers to choose it. My writing on logic and creativity, customer intent and dependable execution in the shop follows the same thread. [S10, S13, S03, S24]

These seven theses connect the book. Each offers something you can test in your own business. The table at the end offers practical signals to watch.





Historical original covers: the 21 January 2025 whitepaper and January 2026 strategy book. Source: agentic-commerce.sh. [S01, S03, S04]

1 · Machine readability is the entry ticket, not the victory

An offer an agent cannot understand does not exist for it. Merchants who bury dimensions, compatibility, delivery terms and prices in images, PDFs or prose drop out before selection begins.

The second half is easily missed. Readability gets an offer onto the longlist. Relevance, availability, price, customer preferences and the agent operator’s rules get it onto the shortlist. A spare-parts merchant with perfectly structured data still loses if delivery takes three weeks.

Treat readability as basic hygiene: you cannot compete without it, but it will not win the sale by itself.

2 · Brands become criteria, while remaining a feeling

The fear that agents will flatten brands underestimates customers. Someone who says “only from this manufacturer” or “prefer repairable, even if it costs more” gives the agent a preference shaped by the brand. The agent carries out a preference formed much earlier: in a shop, through a recommendation or in use.

Brands face two tasks. Every promise needs a verifiable form: “durable” becomes “a ten-year warranty, with spare parts until 2036”. The places where preference forms also become more valuable. A strong brand in the agent era can be felt emotionally and justified factually.

3 · Gatekeepers change shape

Open interfaces lower the cost of access. They do not, by themselves, redistribute power over that access. Whoever controls the preferred assistant, product index, identity or payment relationship stands at a new gate. In the last decade, gatekeepers were search engines and marketplaces; in the next, they may be assistants, wallets or procurement networks.

Merchants should build several routes to demand: use open protocols, maintain multiple channels and make the direct relationship strong enough that customers explicitly send their agents to them. Chapter 19 shows how to prepare a second route in practice.

4 · Trust needs evidence rather than a universal score

A single trust score is convenient and usually wrong. An agent buying for a hospital weighs delivery reliability differently from one buying a gift. Useful signals have a source and a period: delivery performance over the past 90 days, common return reasons or documented repair times.

Merchants should maintain and publish these signals themselves rather than wait for a third party’s score. A number without a verifiable basis is decoration with decimal places.

5 · As routine shrinks, experience matters more

Agents take over work people rarely enjoy: reordering printer paper, comparing tariffs or redirecting deliveries. That makes the time customers willingly spend with a brand more valuable: trying an instrument, planning a kitchen or browsing a collection.

The experience moves too. It unfolds in an assistant’s conversation, a browser agent or an interface the merchant does not control. How product knowledge and brand expression reach those places is one of the most interesting open questions in this field; we explore it through the AXP prototype in Chapter 6. My test for a good experience is simple: it should make a decision safer, faster or more enjoyable. [S08, S09, S12]

6 · The shop becomes an execution layer

Agents become the purchasing interface; the shop beneath becomes the execution layer. A conversation can prepare a request. Quantity, price, tax and order become binding in the commerce core, where business rules apply and someone takes responsibility for them. [S22, S24]

Protocols such as UCP, ACP and MCP are the languages between these worlds. They are essential, but do not provide the grammar underneath: who may grant a discount, when an offer is valid or what happens after a timeout. “We support UCP” is therefore only half an answer. The other half is the version, supported capabilities and purchasing journeys actually tested.

7 · Openness makes individuality affordable

For two decades, commerce followed a simple equation: standard was cheap and distinctive was expensive. Many good merchant ideas failed because implementation cost too much. Coding agents such as Codex and Claude Code change that equation. When merchants or agencies can put an agent to work on open code, “that would be nice” can become a functioning prototype within days.

Open interfaces also allow changes on closed platforms. An open core goes further: agents can read the relationships, draft changes and let people examine them through to execution. I therefore expect open source to grow in importance during the agent era. Agencies gain an opportunity to deliver more of the work merchants previously had to put off because of cost. Routine development takes less time; business understanding, architecture, review and dependable operations determine the value of the solution. Part V explores how merchants and partners can use this freedom.

Seven theses in daily business

A thesis becomes useful when it changes a decision. Each of the seven can be followed through specific observations in the business. The last column describes developments that would weaken the relevant assumption.

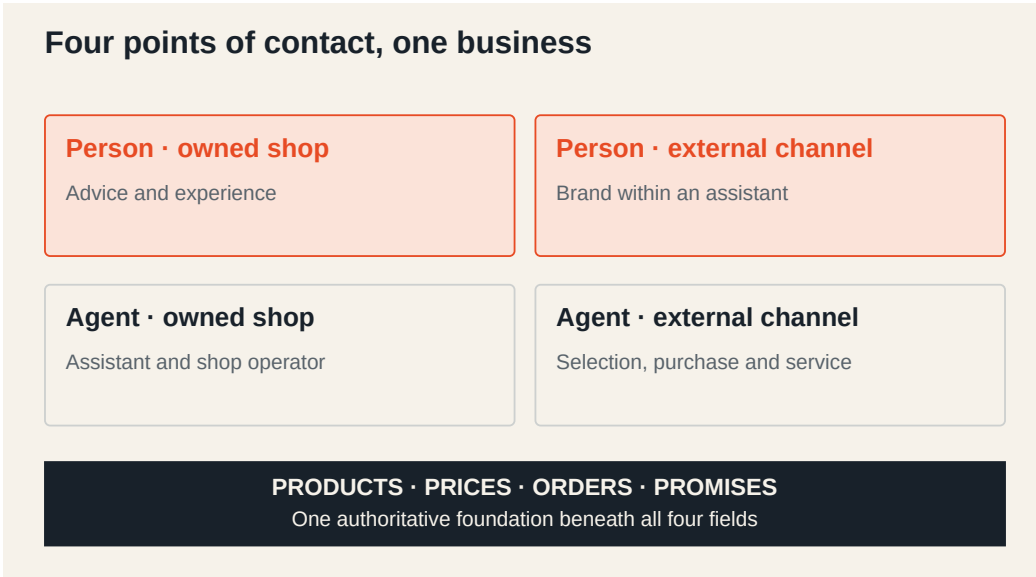
Thesis	Observation in the business	What would count against the thesis
1 Readability	Are the top 50 products identified correctly?	Agents consistently choose correctly despite missing or contradictory product data.
2 Brand	Customers specify brands, origin or repairability.	Price alone determines selection among comparable offers over a sustained period.
3 Gatekeepers	Demand share and fees change by intermediary.	Access remains spread across more providers while intermediation fees fall.
4 Trust	Delivery and service evidence affects selection.	One trust score produces equally good decisions in different purchasing situations.
5 Experience	Advice changes selection, confidence and returns.	Full delegation delivers the same satisfaction for purchases customers value experiencing.
6 Execution	Offers and orders use the same business rules.	Agent platforms take over the entire binding commercial logic.
7 Openness	Distinctive adaptations become less costly to operate.	Closed systems permit the same changes with equal control and total cost.

A spare-parts merchant may follow readability and execution first; an interiors business may focus on brand and experience. The next chapter follows our traveller again: what does she do herself, what does her agent handle, and where does the merchant come in?

II

Customers, brands and experience

The traveller arrives with something to accomplish. She can search, seek advice or send an assistant. This part follows her selection: how offers are found, how brands matter and how an experience helps a decision.



CHAPTER 04

The new customer journey



One journey, several hands. The request travels with it.

A journey with a new division of work

The traveller has made her request. She lets the assistant search, takes a closer look at two suitcases and asks an adviser about the wheels. She decides how much to delegate. This division of work shapes the new customer journey: who handles each step, and how does the next person or system pick up the choices already made?

The adviser explains which suitcase rolls more quietly on uneven ground. The assistant combines that advice with budget and delivery requirements; the shop will later handle the order. The adviser’s knowledge helps determine which choice suits the traveller.

Two kinds of intent

A **defined need** has relatively clear success criteria: a known toner cartridge, a specified replacement part, the same coffee as last time. An **open intention** is less precise: make a living room more comfortable, choose a suitable present or improve a team’s equipment.

In the first case, reduce unnecessary interaction; in the second, help the customer develop useful criteria before recommending a product. An agent can ask questions, show examples and explain trade-offs. A premature product recommendation solves the wrong problem if the customer does not yet know what “better” means.

Handoffs as a product capability

A good handoff saves the customer from repeating work. Someone moving from an assistant to personal advice should find their request and selection waiting there. Service needs the requirement, relevant conditions, open questions and next decision.

Some handoffs are a deliberate part of advice: a customer wants to feel a material, discuss a difficult fit or confirm an important purchase with a person. The agent prepares the conversation and passes on the information it needs. Personal service becomes easier to reach and can begin with the question that matters.

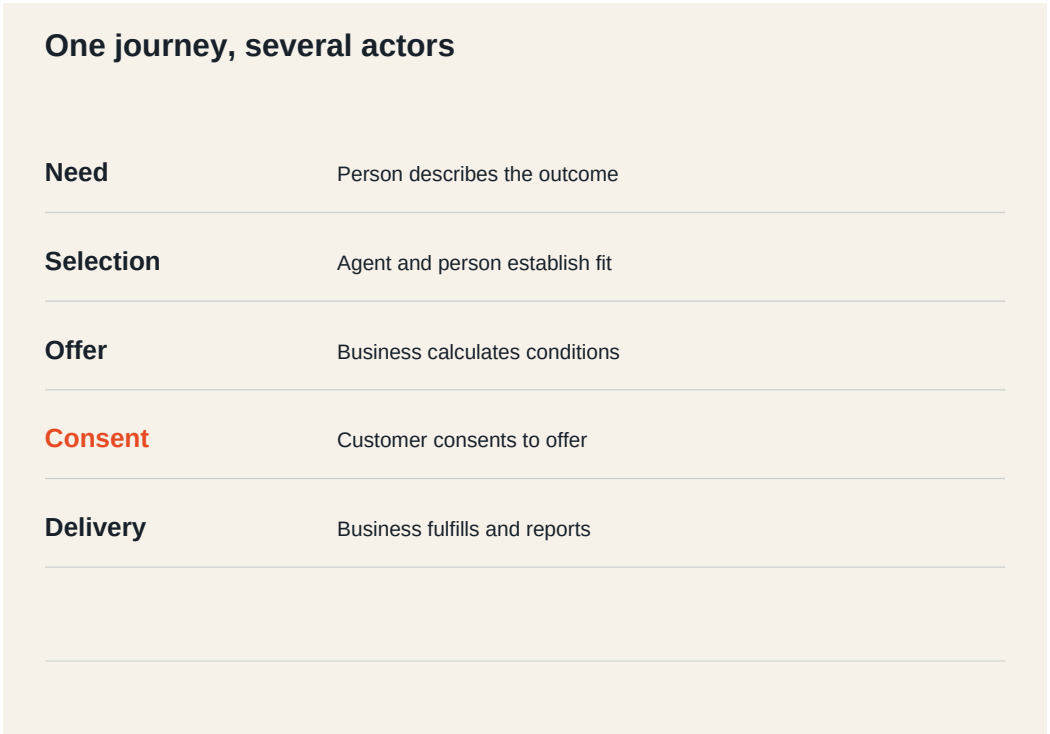


Figure 6. Customer, agent and merchant collaborate along one journey. Good handoffs preserve the request, selection and consent.

Measuring a different kind of success

Good advice helps a customer make the right decision. Sometimes it leads straight to a purchase; sometimes to a tape measure or a fabric sample first. Measure how often customers complete their task, which questions remain and how frequently a selection needs correcting later.

Familiar metrics remain useful when matched to the step: a helpful recommendation, a confirmed basket and delivery without a complaint. Together, they show whether the new route improves things for customers and merchants. A high conversion rate with the wrong products would mainly be an expansion programme for the returns department.

One person owns the whole purchasing journey

Someone in the business needs responsibility for the whole purchasing journey and the quality of its handoffs. Marketing can improve selection, development can improve checkout, and customer service can handle complaints. One person coordinates the handoffs between them.

In a small business, that person may be the managing director. With the advisers and whoever runs the shop, she follows ten frequent requests through to delivery. They discover that the assistant finds the right model but loses the promised delivery date when it opens the basket. An abstract AI project becomes a specific task with someone responsible for it.

The merchant needs three perspectives: someone decides on money and objectives, someone knows the products and the customers' problems, and someone makes sure the selection becomes the right order in the shop. In a small business, two people may cover all three responsibilities.

Responsibility	Task in the suitcase purchase	Visible result
Management and sales	Set the budget and expected benefit	After its costs, the new route contributes to the business.
Product team and advisers	Establish outer dimensions, wheels and airline suitability	The traveller receives a selection she can assess.
Shop operations and development	Connect price, consent and order	The offer the traveller consented to becomes the correct order.

Customer service and accounting later show where the journey creates extra work. Their findings feed into the next change, for example by answering a frequent question before the purchase.



Figure 7. A shared customer task connects commercial responsibility, product knowledge and platform operations.

Work on the journey therefore often starts with employees’ knowledge. They know the size customers choose incorrectly and the delivery promise that causes trouble every Friday. That knowledge must also reach the place where a customer asks an assistant for a suitcase.

CHAPTER 05

Being found before anyone visits the shop



If you want to be found, do not hide your wheels.

Being found is only the beginning

The suitcase looks right, costs €169 and is available. One detail is missing: do the dimensions include the wheels and handle? Without it, the assistant cannot check the cabin-baggage rule. The merchant could have exactly the right product and still miss the shortlist. That is where discovery in an agent market begins: an offer needs to be found, understood and recognised as suitable for the task.

It would therefore be wrong to declare search optimisation universally obsolete. Clean URLs, accessible pages, understandable content and accurate structured data remain useful. The new requirement is to connect those foundations with actionable offers and verifiable conditions. A mention does not create a deliverable basket. [S14, S17, S18]

An assistant builds a shortlist from sources it can access and trusts. These may include its own product index, merchant feeds, marketplaces or direct shop connections. Their operators help determine which offers the customer sees.

The spare-parts merchant can repeat frequent sales enquiries in the assistants its customers use. The same items reveal whether selection, variants and delivery dates arrive correctly. Missing details can then be traced to the relevant product page or feed.

Brand preference often forms earlier

Measuring branding only at the last touchpoint misses its earlier influence. A customer can tell an agent, “I trust this brand, avoid that one, and will pay more for repairability.” The agent then implements preferences formed through experience, communication or community.

The promise has two forms. People experience materials, origin and repair service as part of the brand. The agent needs the details that let it apply that preference during selection: material composition, warranty and spare-parts provision.

The route can be followed through three results: the product appears for a relevant request, the assistant explains it accurately, and the customer reaches a corresponding offer. A fixed set of typical customer requests reveals changes over time.

The spare-parts merchant distinguishes explicit brand searches, independent recommendations and paid placements. This shows its marketing team which demand the brand generates itself and where another entry point earns its cost.

The information that makes the difference

For the suitcase, useful improvements start with clear product identities, complete dimensions, understandable delivery terms and consistent prices. Files such as `llms.txt` or a public `AGENTS.md` can complement that foundation where the intended systems actually use them.

The most valuable information often answers a customer question that previously remained open. For the suitcase, that might be external dimensions including wheels and handle. Maintaining that detail helps an agent as much as it helps a person.

Advertising inside the answer

When an agent's answer determines the shortlist, that decision point becomes attractive to marketing budgets. On 16 September 2026, OpenAI announced tests of Sponsored Agents: after clicking an ad, users can talk to a business-sponsored agent. Paid entry into a conversation is becoming a concrete distribution approach. Budget owners need to distinguish organic selection, paid placement and intermediary compensation. [S167]

The merchant agrees with its marketing partner how advertising is labelled and what the buyer agent discloses to its principal. Paid placement must still satisfy the hard criteria: sponsorship does not make a €230 suitcase suitable for a €180 budget.

The commercial test is the additional contribution after fees and the costs of returns and service. High conversion can also arise because an assistant receives customers already ready to buy. What matters is how much business would not have happened without the spending; Chapter 14 develops the calculation.

A user-agent name provides a clue but can be imitated, so the merchant distinguishes verified agent access, suspected automation and unknown traffic. Signed requests or authenticated interfaces identify a sender more reliably than browser characteristics; the customer task and permission to buy still require their own evidence.

For analysis, the merchant connects requests, offer revisions, baskets and orders through identifiers recorded by the shop. Some assistants refer visitors to a shop; others purchase through an interface without creating a normal web visit. Web analytics alone misses the second route. The merchant collects the necessary information for a defined purpose and retention period, distinguishing estimates from verified transactions.

Block agents or invite them in?

Merchants need protection against excessive requests while remaining accessible to customers. A blanket ban on AI access combines different interests: model training, product search and a specific customer task serve different purposes.

OpenAI, for example, distinguishes GPTBot for potential model training, OAI-SearchBot for search and ChatGPT-User for user-initiated visits. Search and training settings can be managed separately; the documentation checked on 15 October 2026 says robots.txt rules may not apply to user-initiated ChatGPT-User visits. Refusing training therefore need not mean refusing discovery. Check each provider's rules, because robots.txt controls neither every request nor permission to buy. [S154]

My recommendation is to make public product information deliberately accessible, limit expensive requests and offer protected access for business actions. A spare-parts merchant can open its catalogue to search while requiring authentication for customer prices. Volume and rate limits control bulk requests; baskets, orders and payments each receive appropriate permissions. A security problem at one entry point can then be addressed there without making every product page invisible.

The initial review distinguishes requests that help customers, requests that merely consume resources and attempts to retrieve confidential information. Verified origin, purpose and observed behaviour provide a better basis than a browser-header name. From that evidence, the merchant can develop a rule, try it against actual requests and adjust it.

Access becomes a commercial decision about bringing suitable demand through search and fulfilling authorised tasks through transactional connections. Reducing requests alone can also reduce reach, while opening everything may create costs without customer value. A useful access policy weighs those effects separately when setting the rules.

Which access serves which purpose?

Training	**Choose deliberately**	Provider crawler rules
Search	**Open public offers**	Verify origin, limit requests
Customer visit	**Support the customer task**	Control session and load
Transaction	**Authorize protected action**	Check identity, mandate, amount

Figure 8. Four purposes need different access rules. These are strategic recommendations; OpenAI says robots.txt rules may not apply to ChatGPT-User. Source S154.

The physical store becomes a promise that can be fulfilled

“Available for collection nearby today” may matter more than the lowest price. A store-based merchant has something a product index cannot create: stock in the right place, advice and service on the same day.

An agent needs branch inventory, opening hours, reservation rules and a dependable collection time to recognise that advantage. The traveller could collect her suitcase on Thursday on her way home. A reservation turns “probably in store” into a usable offer, and staff must be able to retrieve that same reservation. A friendly conversation with an agent does not put a suitcase behind the counter.

For omnichannel merchants, the first investment connects the local promise to actual stock. Advice, repairs and in-store returns can strengthen the offer. By making these services easier to compare, agents can also reveal the distinctive value of a physical store.

From catalogue to decision

A catalogue describes a product, while customers want to know whether it solves their problem: whether the seal fits the machine, the lamp lights the room or the suitcase arrives before departure. An agent needs those same answers to make a useful recommendation.

The details matter because two manufacturers may use the same article number, while “waterproof” needs defined operating conditions. “Available” also remains ambiguous until it is clear whether stock is on hand or must first be ordered from a supplier. Useful product data resolves these ambiguities.

Information	The customer’s question	Example
Product	Is this the exact model I need?	Part number, variant, revision
Suitability	Does it fit what I want to do?	Dimensions, material, compatibility
Offer	Which terms apply to me?	Price, tax, volume discount
Availability	When can I get it?	Quantity, stock location, delivery date
Service	What happens after purchase?	Shipping, returns, repair
Evidence	What can I rely on?	Source, test date, certificate

Much of this information already exists in the business: in operational systems, product management, the shop or customer service. The shared task is to make it useful where a customer or agent makes a decision.

Some information remains stable; other details change constantly. A suitcase's material can be maintained at a steady pace. Price, stock and delivery need to be current before purchase. The shop therefore recalculates the specific offer when selection becomes an order.

The suitcase merchant assigns a responsible source and an appropriate update interval to each decisive fact. This directs effort towards information that must be correct at the moment of purchase. If a condition changes, the assistant explains the effect on the task and seeks fresh consent where required.

Good advice recognises when to ask a question. If a spare part's fit is uncertain, the machine revision or a drawing may resolve it. Missing allergen information calls for a confirmed manufacturer response.

The spare-parts merchant distinguishes verified facts from unresolved points so the agent can ask a useful question. Uncertainty becomes a practical step towards the right choice.

Quality work with an economic focus

The spare-parts merchant starts with an important product group and asks service staff which missing facts regularly cause questions or wrong orders. It assigns sources to those attributes and reconciles the product page, feed and checkout. Data work first improves decisions that actually burden the business.

Completeness alone says little about that result because a hundred populated fields may help less than one dependable compatibility rule. The merchant also tracks incorrect recommendations, necessary questions and subsequent product corrections.

Practical task. Select twenty representative products. Ask purchasing, sales and service separately to explain when a customer should not buy them. Differences between their answers often reveal the most valuable gaps in the data model.

The suitcase can now be found. Whether the traveller also wants it is decided elsewhere.

CHAPTER 06

Experience that improves a decision



A grand stage. And a suitcase that still needs to fit.

When shopping deserves attention

An assistant presents a sofa as a picture, a price and three bullet points. The customer sees the product, but little of what a good merchant adds: fabric samples, a configurator and the knowledge that pale upholstery and cats make an ambitious combination. Purchases like this show why experience matters as agents take on more routine work. People want to feel confident about their choice and enjoy making it.

A brand can serve customers in two ways: make delegation dependable and create reasons for voluntary attention. Automated replenishment and an inspiring store visit can belong to the same relationship. Expectations and operating processes must support each other.

A gift should express something, an instrument must feel right and a sofa needs to fit the life of a room. A list of attributes rarely suffices because experience, taste and

personal advice create part of the value. I describe these situations as emotional value zones. [S03, S10, S12]

A room view helps customers understand proportions, while the question they bring determines whether that is the useful medium. A material video shows a difference that a photograph struggles to convey. For two technical variants, a clear comparison can provide the best advice. Good experience chooses the form that makes the decision easier.

In my adaptive-storefront prototypes, presentation follows customer intent while product and transaction data comes from the controlled commerce core. I pursue this division of work because it gives design freedom while keeping purchases on dependable foundations. [S09, S08]

A generated image can convey atmosphere and use while fixed components supply price, variant selection and availability. Both must match the article customers will actually receive.

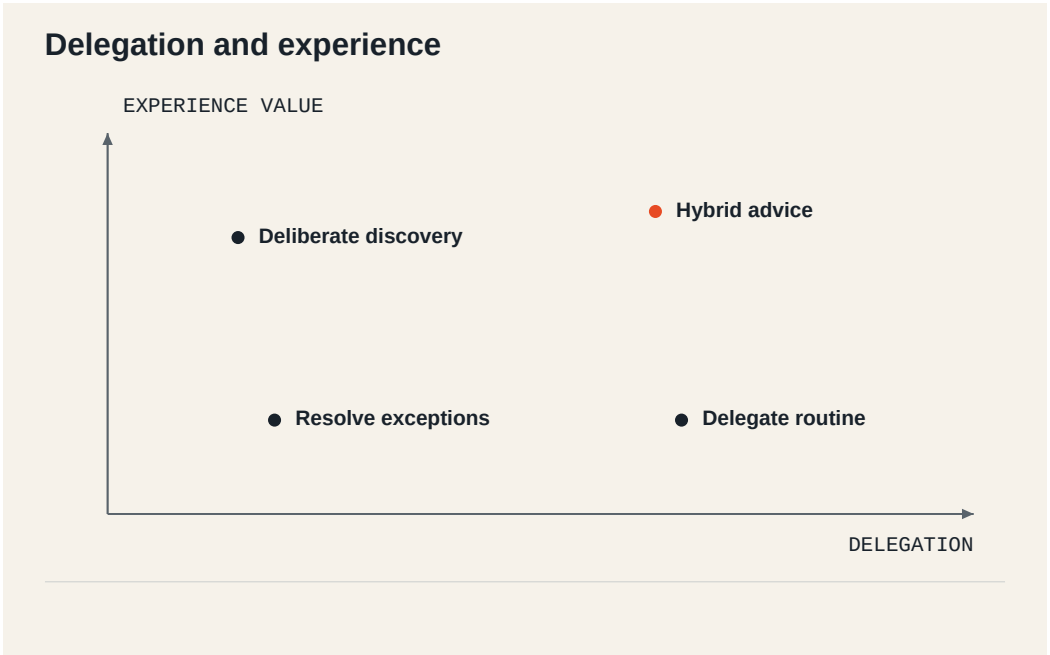


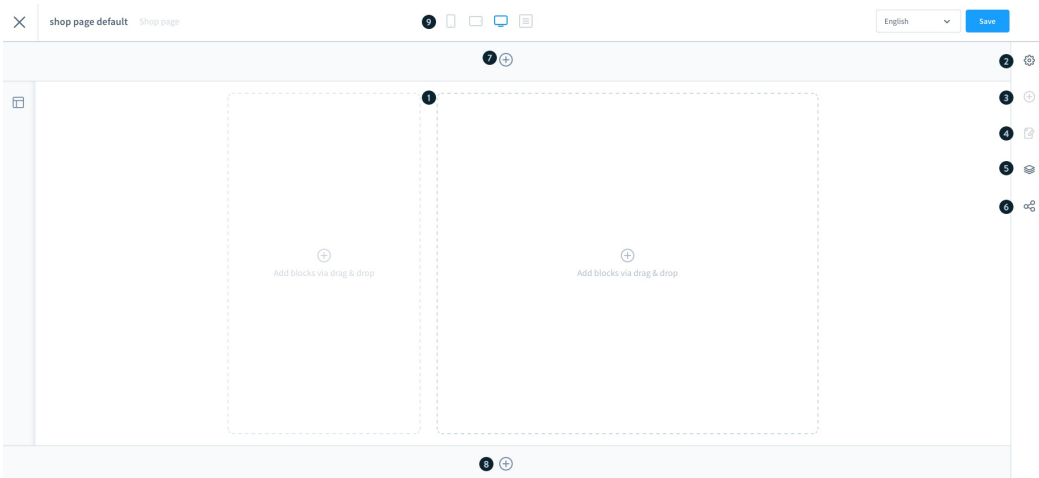
Figure 9. Strategic framework for delegation and experience. One purchase can connect several areas.

Experience becomes a response to intent

For a long time, a store was mainly a collection of pages. A homepage had to address many people at once; a category had to accommodate many different intentions. That was a sensible response to the technology available. It is not the only form digital commerce can take.

A useful experience starts with the occasion. Someone choosing a coffee machine for a small office needs different guidance from someone perfecting an espresso ritual. Product, imagery, explanation, comparison and advice should reflect that difference. As systems handle more background work, there is greater freedom to organise the visible experience around the customer's actual decision.

At Shopware, connecting content and commerce has long been part of the product idea. Shopping Experiences provides a system of sections, blocks and elements for landing pages, categories and product pages, with content connected to product data. It gives design a reusable foundation rather than requiring each campaign to start from scratch. [S121]



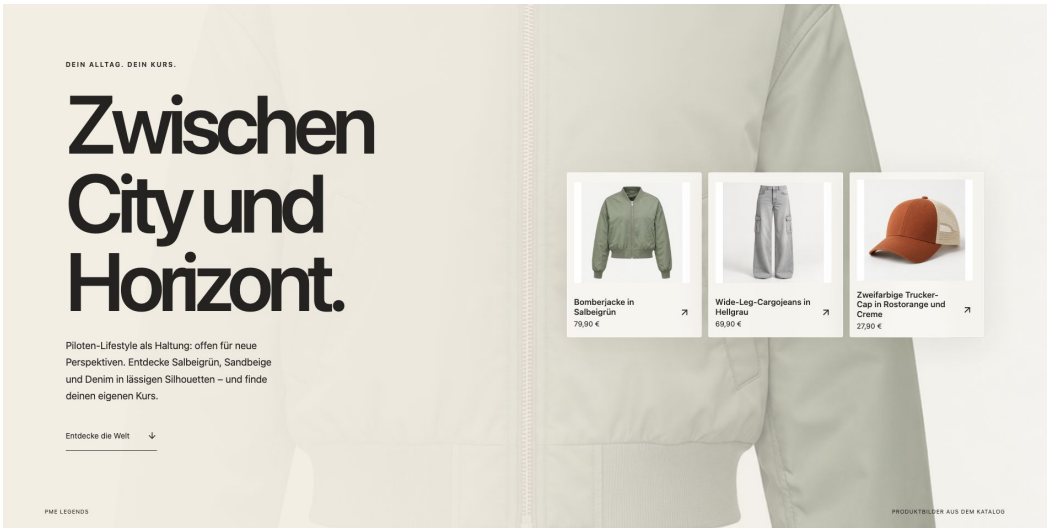
Shopping Experiences: original layout-editor image from Shopware documentation. Sections, blocks and elements connect design with commerce content. Source: Shopware. [S121]

A team can describe advice or a product story and develop several designs with agents, while the brand directs atmosphere, selection and claims. An office coffee machine calls for easy operation; an espresso ritual calls for personal preference. Good design makes that distinction visible and supports the particular decision.

Storyfront: from a product page to a product story

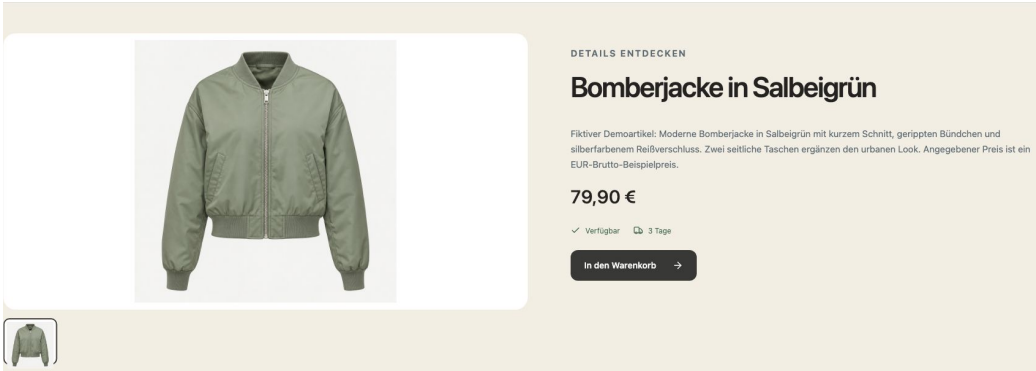
Storyfront explores this transition. A product selection can become a connected story, combining spatial representation, movement, materials, explanation and advice that responds to the current question. Customers need not learn a company’s internal catalogue structure before making progress. The presentation can develop around the task they brought with them.

For a lamp, the designed experience can begin with its effect in a room and explain light, material and size along the way. Selecting a variant brings the binding purchase process into view. A technical product may instead benefit from an explanatory assembly with operating conditions and configuration, because the medium should help people understand their actual need.



Storyfront demo from my VEYNDO experimental environment: a fashion experience with directly linked products. Original capture, 14 October 2026; German interface. [S129]

In the fashion demo, a jacket, jeans and cap form a coherent selection with a distinctive visual language. Clicking the jacket opens details with price, availability and purchase options, turning a mood into a traceable route to the article. Presentation changes while the underlying item remains the same. [S129]



From the editorial opening to a specific jacket, price and purchase option. Storyfront demo from my VEYNDO experimental environment; German interface. [S129]

The publicly documented Storyfront integration sends product identities and quantities through the same basket and ordering operations as the native interface. Because the commerce core calculates price, tax, shipping and availability and returns the confirmed outcome, the designed experience can offer greater freedom of presentation while keeping purchases dependable. [S126, S127]

The designed experience creates meaning while the commerce core makes commitments binding. A material story can therefore be emotional, provided the purchase still contains the correct variant.

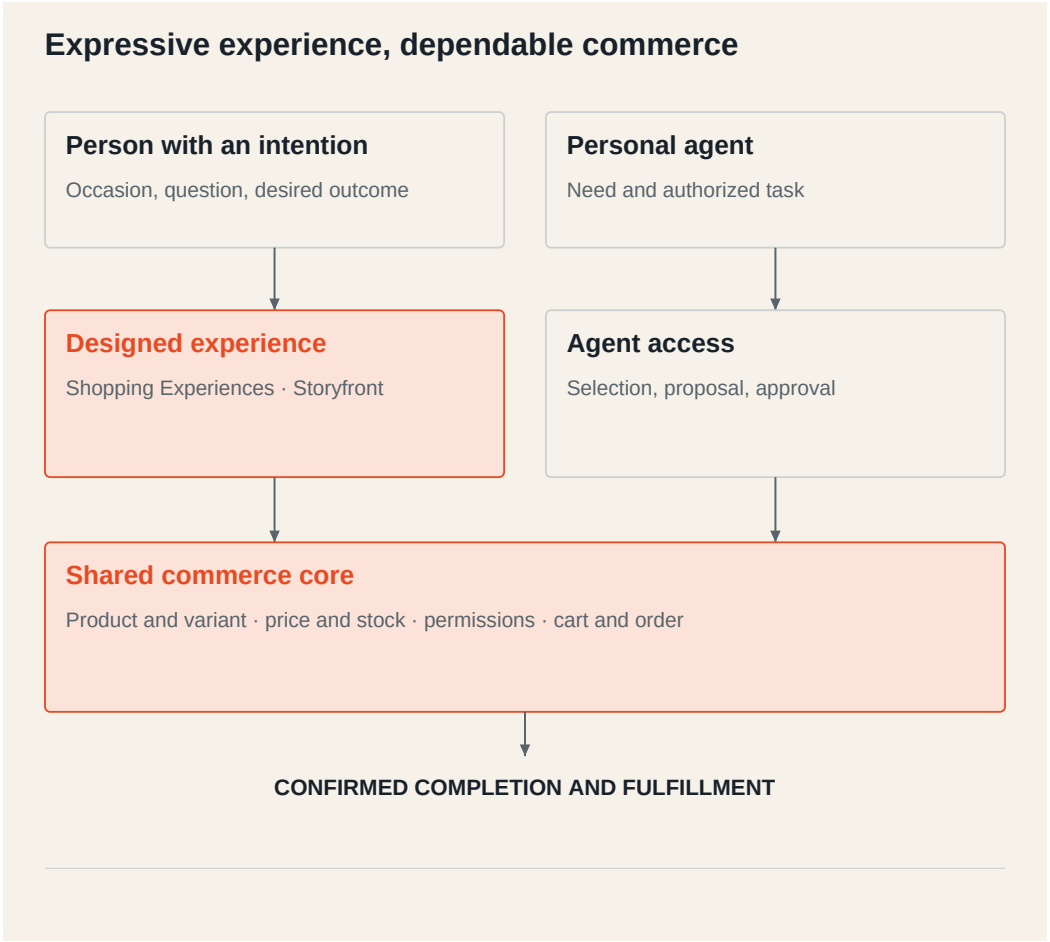


Figure 10. Design model based on Shopping Experiences and the documented Storyfront integration. Interfaces share authoritative commerce operations. Sources S121, S126 and S127.

From my experimental work · Storyfront

Task. Connect a product story to a selection that can be ordered.

Setup. A fashion demo with linked product details and integration with shared commerce operations.

Finding. Presentation can vary while product identity, variant and basket remain consistent.

Status. Demo with a test catalogue and simulated payments. Next: measure how the presentation affects selection in merchant operations. [S126, S127, S129]

Lichtfeld (fictional): generative lighting advice

Lichtfeld sells lighting and helps with room design. A customer wants a more welcoming living room and brings photographs, dimensions and an idea of the desired atmosphere. The combination should suit the room and existing furniture.

The assistant asks where people read, work or relax. Room views make different solutions imaginable. Alongside them are orderable products with dimensions, colours and installation requirements. Each decision turns the atmosphere into a specific selection.

The customer may prefer a light whose chosen colour is temporarily unavailable in their country. Advice can offer an available variant or another delivery date. If the customer wants to speak to a person, the adviser receives the requirements and selection already made. Planning continues even as the interface changes.

During testing, the team disables the room view and continues through images, specifications and conversation. It checks that the displayed variant appears in the basket and that the customer understands its actual proportions. Acceptance connects design with the product that will arrive.

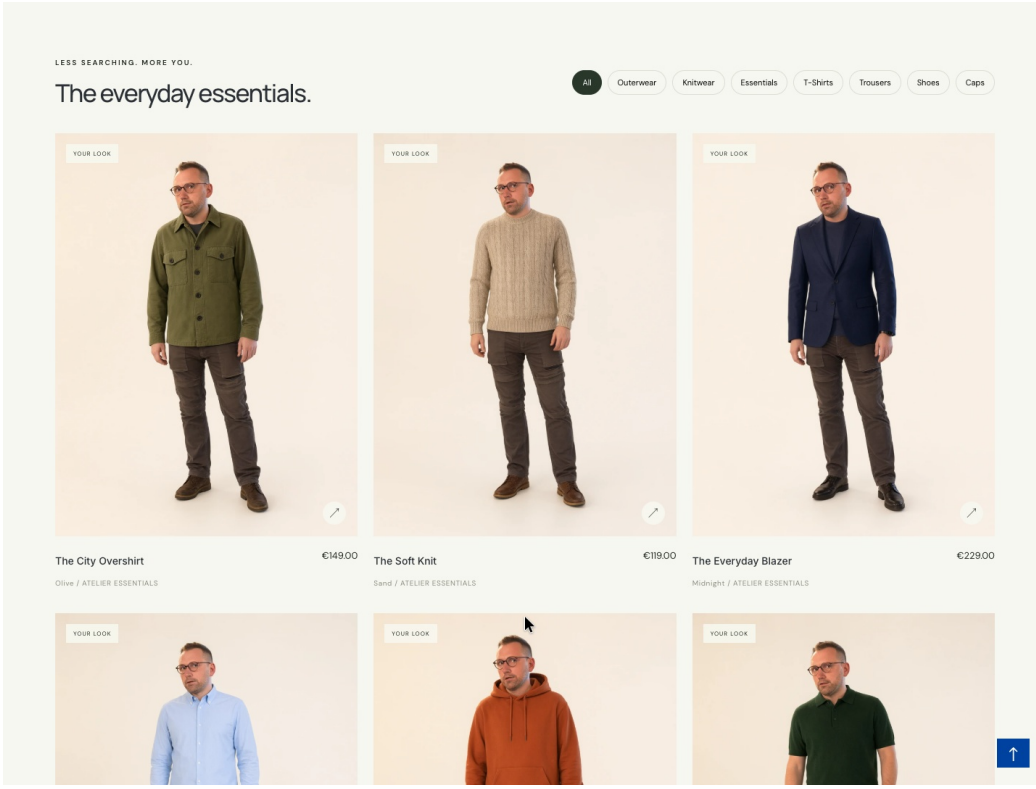
Lichtfeld looks for better decisions: greater confidence, suitable combinations and fewer avoidable returns. Content design and maintenance are assessed against those effects. The business can see where a distinctive experience also makes commercial sense.

Shopware’s public Community Day 2026 architecture connects Commerce Core, Nexus, Copilot, Payments and Experience Studio, introduced as a research preview. The core retains business logic, Nexus brings context together, Copilot works with it, and Experience Studio develops expressive commerce interfaces. The direction matters: more operational intelligence and more creative freedom on one foundation. [S123]

For me, these developments belong together. As an agent takes on background work, the person in front of the screen need not receive an increasingly generic interface. The capacity released can support better advice, more personality and a stronger individual identity. Automated operations and distinctive experience are two sides of the same opportunity.

ATELIER / YOU: turning imagination into a personal catalogue

My ATELIER / YOU experiment turns a reference photograph into a personal view of clothing and outfits. The Shopware storefront connects an image service to the session and product combination. I am interested in how closely imagination and selection can converge when presentation becomes personal while remaining tied to the product. [S113]



ATELIER / YOU: personal product views from a reference photograph. Original screenshot with AI-generated demo content. Source: Stefan Hamann. [S113]

From my experimental work · ATELIER / YOU

Task. Visualize clothing using personal image references.

Setup. An evaluation project with personal image generation and a synthetic catalogue.

Finding. Visual presentation and fixed product facts need separate treatment.

Status. Prototype for personal product presentation without an ordering flow. Next: examine image fidelity, usage rights and effects on selection and returns. [S113]

When experience moves out: the AXP prototype

Return to the brief sofa card at the start of this chapter. Fabric samples, a configurator and advice need a place in the outside assistant too when they help the customer choose. Part of the merchant's experience needs to travel with the customer interface.

That gap motivates AXP, the Agentic Experience Protocol prototype from our Agentic Commerce Lab. Its research question is how merchants can bring richer product knowledge and experience into an assistant's interface. [S50]

The prototype explores three answers: richer product types for variants, bundles, subscriptions and configurators; quality signals with provenance; and embedded experiences such as 3D or AR components communicating through defined events. Binding purchasing remains on the commerce route, for example a UCP checkout session.

For me, the most interesting issue is brand expression. A brand wants to express tone, imagery and a point of view. The assistant wants a useful answer that fits the conversation. A new balance has to emerge between the two, much as it did between brand shops and marketplaces, except that this time the assistant sets the rules of the stage.

MCP Apps raises related questions about interactive interfaces, host support and security. We need to understand which experience components travel well, how the host isolates them and what customers gain. A merchant should begin with structured facts and useful comparison signals, then add interaction where a channel supports it. The early layers remain valuable whichever approach gains adoption. [S67]

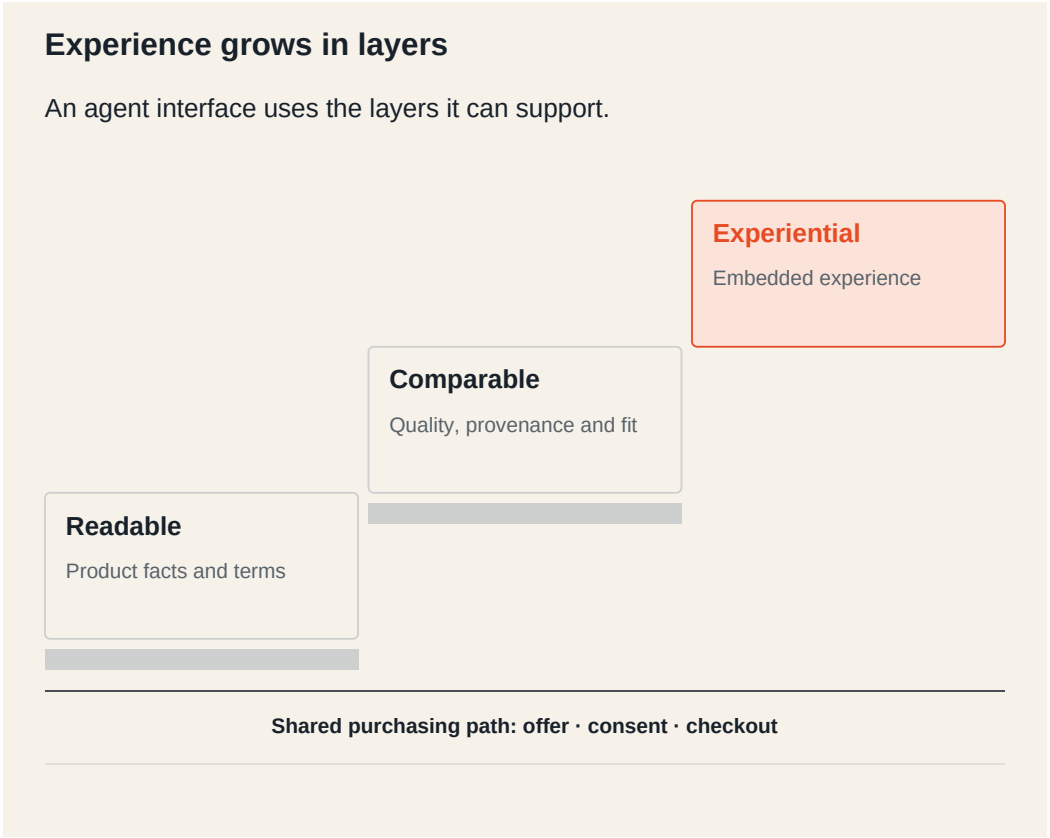


Figure 11. Experience travels in layers. Each layer remains useful without the next; host support determines how far it can go.

From the Lab · AXP

Task. Bring product knowledge and interactive advice into external assistants.

Setup. A research prototype describing product data, quality signals and embedded components.

Finding. Layered presentation connects usable facts with optional interaction.

Status. Schema draft dated 13 January 2026. Next: work with hosts to test which embedded elements they support and how much brand expression they allow. [S50]

After planning a room, a customer may be able to say exactly which light she wants. She understands its proportions, has seen the material and knows how it is installed. She can later give that knowledge to her personal agent. The experience influences choices beyond the visit to the shop.

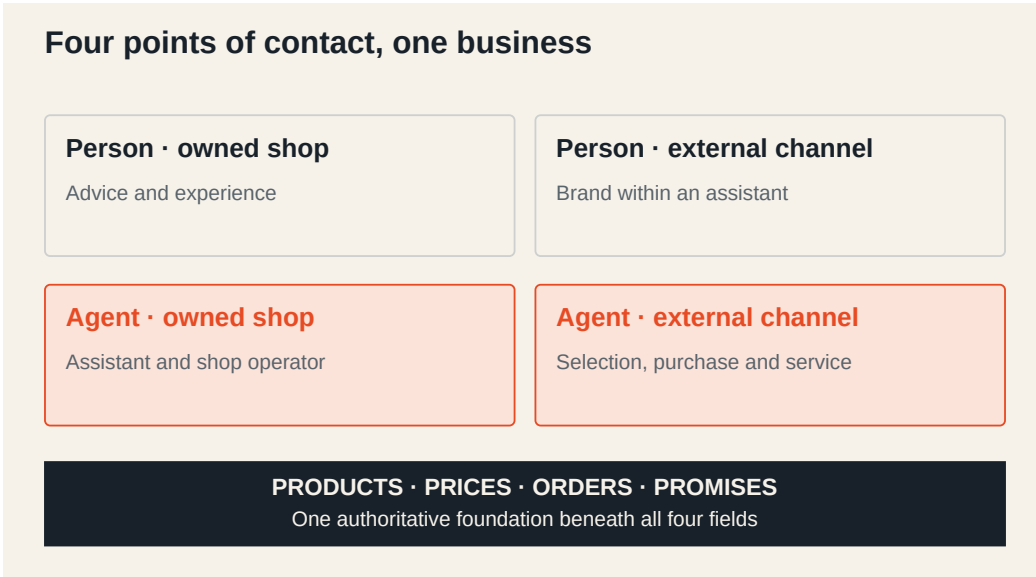
The merchant assesses a designed experience through fewer sizing errors, fewer installation questions and greater confidence about combinations, weighed against design and maintenance costs. Generated design is easy to reproduce, while the necessary product knowledge and advice develop within the business. Chapter 20 builds differentiation on that foundation.

The experience then needs a route to purchase. In the next part, we follow the suitcase from request through offer to delivery and place the protocols alongside the tasks they perform.

III

The purchasing journey and its protocols

The suitcase now needs to arrive by Friday. Agents, shop and payment provider must agree on the item, terms and approval. We follow the purchase through to service and place protocols alongside the tasks they perform.



CHAPTER 07

How protocols become a purchasing journey



Many connections. One shared destination.

One suitcase, several participants

The traveller wants a suitcase, not a networking project. Yet several parties need to work together behind that simple request. Who knows which models fit? Who calculates delivery? Who may place the order? Who reports a delay? Protocols organise that cooperation. Their names become easier to understand when we follow the work they do along the purchase.

For the purchasing journey to work, participants need a shared understanding of the product sought, the applicable price, consent and order status. Protocols give that exchange a form, much as agreed measurements, order forms and delivery records do in established trade.

That explains the apparent protocol zoo. The names represent different jobs along the same purchasing journey. Some help with discovery, others with access to business

functions, purchasing or payment. Understanding the journey makes the initials much easier to place.

The journey provides the map

The **customer's intent** leads to a **selection**, for which the merchant calculates an **offer**. Following the traveller's **consent**, order and payment are **executed**, followed by **delivery and service**. This familiar purchasing journey shows the job each protocol performs.

Three things need to travel through the whole journey: product information, the customer's permission and the progress of the task. For our traveller, it is still the same suitcase, within the same €180 budget, arriving before the same trip. Moving from a chat to a shop should preserve that context.

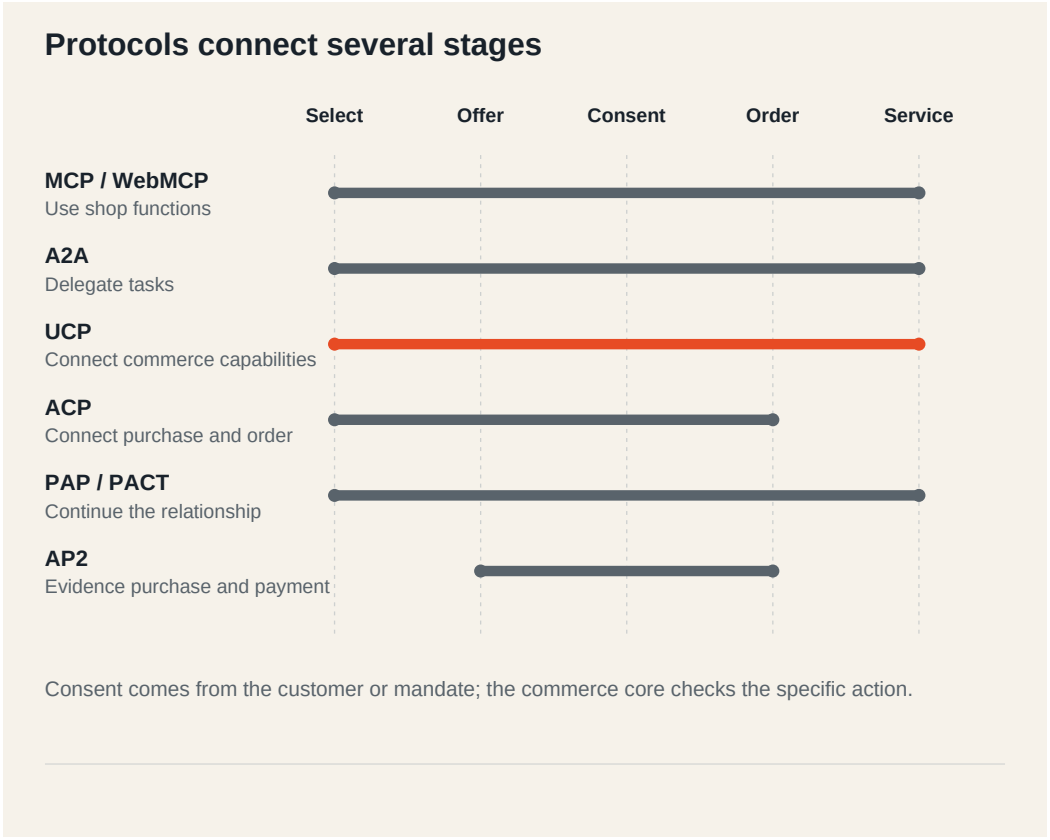


Figure 12. Working map of possible roles along a purchase. Bars indicate task areas, not guaranteed integration capabilities. Check scope and revision in the protocol passport. Sources S28–S32, S34, S38.

The approaches shown in the diagram are different ways to perform the jobs. A merchant connects the components its customers and partners use. One protocol may

cover several tasks: UCP, for example, spans the discovery of commerce capabilities through to order events, while MCP can provide access to those same functions. [S28, S30]

Today, a merchant often needs a separate integration for each partner. The same basket is described differently, delivery terms go into different fields, and status messages vary. Common protocols reduce this translation work. A new interface can connect more easily to existing commerce functions.

The division of work remains familiar: merchants manage products, prices and orders while assistants help customers navigate them and payment providers handle money movement. A protocol makes the connections between those responsibilities reusable across purchasing journeys.

MCP gives the agent tools

Suppose the assistant wants to know which suitcases meet the cabin-baggage requirements. It needs a search function and then accurate variant information. The Model Context Protocol, or MCP, describes a shared way to make tools and data available to AI applications. An application can use it to search products, check stock or prepare a basket. [S30, S31]

Existing functions can continue to do the work: product search uses the catalogue and prices come from the shop’s pricing logic. MCP makes those foundations available to another user rather than requiring the merchant to rebuild them for the agent.

An employee understands the difference between checking a price, preparing an offer and submitting an order. An agent needs the same distinctions in its tools. Clear tasks help it choose the right next step and let the merchant see which step the agent is carrying out.

A useful response might say: “The suitcase is available. This offer is valid until 4 p.m. The customer’s consent is still needed to place the order.” The assistant can work with that. When something goes wrong, the response should also explain the next step: obtain a new offer, find an alternative or involve customer service.

The shop checks the required permissions. Product search and a binding order remain separate activities, as they are in daily operations. That separation creates room to delegate more work over time.

A2A connects independent partners

A direct stock query returns a number immediately, whereas coordination with another partner can take time and require questions. If the suitcase needs delivery to a hotel, a logistics partner may take responsibility for that part of the task and return with an answer later.

Agent2Agent, or A2A, describes messages, tasks and results exchanged between independent agent systems. An Agent Card introduces the work a partner can undertake and how to reach it. The requesting agent can bring in support without knowing how the partner works internally. [S32]

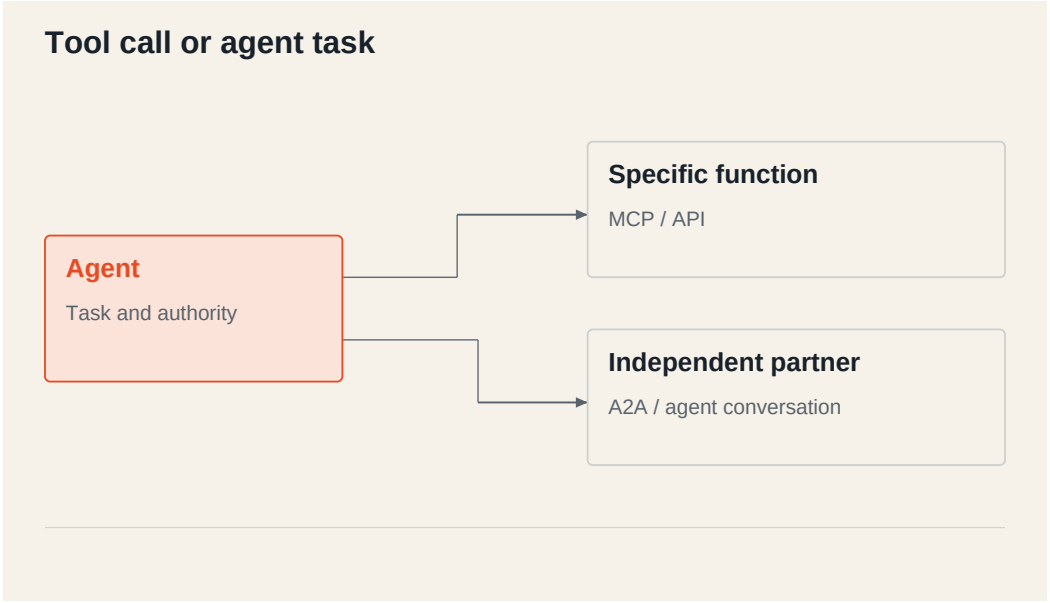


Figure 13. An agent uses a specific shop function or assigns a task to an independent partner.

The traveller wants to explain her request once. When an agent passes on a delivery question, the deadline, destination and permitted choices should travel with it. The partner should receive only the information needed for its job. Hotel delivery requires an address, rather than a copy of every conversation about the trip.

The delivery partner acknowledges the assignment, sends a follow-up question and later returns a result. The buyer agent knows whether to wait for an answer or prepare the next step. The traveller receives the hotel-delivery information without coordinating the exchange herself.

Several agents are particularly useful when responsibilities differ or tasks can run in parallel. The arrangement should make purchasing easier. Five agents taking turns to summarise the same catalogue look more like a meeting with a higher electricity bill.

The traveller can now receive an offer. The next step turns it into a purchase she has consented to.

CHAPTER 08

From the offer to the service that follows



The purchase is one stop. The relationship travels on.

Now the suitcase needs to arrive

The traveller has chosen one of three suitable suitcases. Search showed a price of €169. Now she needs the complete offer: the right variant, a price including delivery and arrival on Thursday. Only then can she decide whether to buy. She expects her assistant to keep that purchase together through delivery.

This is where commerce protocols help. They describe the purchase so that an assistant and a merchant can work on the same transaction. The customer can stay in the conversation while the commerce core calculates the terms and maintains the order.

The offer records the decision

The offer records a total including shipping and tax, the correct variant and delivery to the chosen destination. The shop calculates those terms and records how long they remain valid.

The customer consents to that offer. If delivery moves from Thursday to Saturday, she needs to decide again even if the price stays the same. A suitcase arriving on Saturday is no help to a traveller leaving on Friday. The delivery date belongs in the offer as firmly as the price.

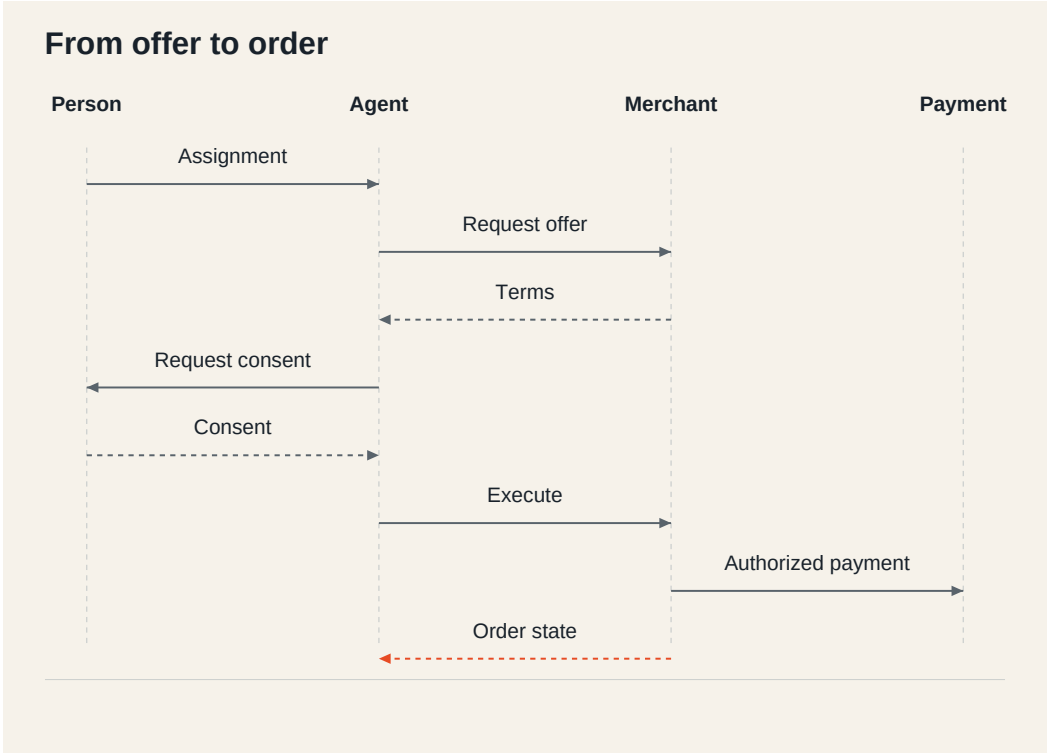


Figure 14. Purchasing as a shared process involving customer, agent, merchant and payment provider.

The customer should see at a glance what she is consenting to: product, variant, total and delivery date. For replenishment, she may have agreed a defined scope in advance. Her assistant might reorder the same consumables up to a monthly spending limit.

This pattern runs throughout the book. People decide what matters to them; systems stay within the agreed scope. Familiar, well-defined tasks allow more to be arranged beforehand. A new financing agreement or a substantial change brings the customer back into the decision.

UCP connects the stages of the purchase

The Universal Commerce Protocol, or UCP, establishes shared terms for collaboration between customer interfaces, businesses and payment providers. A merchant describes the commerce functions it offers. Depending on the supported scope, these include product search, baskets, checkout, account linking and order information. [S28]

For our traveller, the assistant might find a suitcase, add suitable accessories to a basket and request a specific offer. After purchase, it can receive information about the order. The merchant presents its capabilities in a common form that different supporting interfaces can use.

> *“a new open standard for agentic commerce” — Vidhya Srinivasan, Google, 2026. [S27]*

UCP allows both sides to identify the functions and versions they can use together. This supports ongoing development: a shop can add capabilities while existing connections continue with their agreed scope. The dated versions and payment integration examined for this edition appear in the protocol passport. [S94, S95, S96]

Google launched Universal Cart in the US across Search and Gemini on 15 October 2026. Customers collect products from different merchants and can buy within Google’s interface or transfer items to the merchant’s basket. The merchant remains the seller and service contact. The cart makes a purchase spanning several interfaces tangible. [S163]

Our traveller has three possible routes: take a prepared basket into the shop, confirm a purchase within the assistant, or ask an agent to buy within agreed limits. These routes can coexist. The merchant needs selection, consent, ordering and service to connect coherently.



Figure 15. Three possible purchasing routes that can coexist. An agent's authority is agreed for the particular task. Google examples: sources S163 and S164; mandates: S38.

Google's Cart API transfers a populated basket to the merchant. The documented starting point uses CreateCart for a one-way handover, without ongoing synchronization. Native checkout and payment integration are not required for this transfer. The cart capability dates from UCP version 2026-04-08. The channel's offering also determines which customers a merchant can reach. [S164]

The merchant's own experience remains valuable across these routes. It can deepen advice, reassure customers about their selection and make the transition to purchase easier. Merchants selling within an assistant need their product knowledge and service promises to be understandable there too.

The Agentic Commerce Protocol, or ACP, from OpenAI and Stripe also describes exchanges between agents and merchants. It began with checkout; the specification examined here also covers baskets, product feeds, orders, authentication and MCP access. [S25, S29, S97, S98]

For strategy, the shared direction matters more than a contest between initials: assistants should be able to interact with a merchant through specific commerce operations. UCP and ACP provide different models and ways to connect. A merchant can support several routes using the same pricing and order management underneath.

The customer channel determines which connection is worth building first. Its participation terms and supported markets are part of the decision. A protocol choice becomes a distribution choice: which route reaches our customers, and what can they do there?

Being found, being selected and completing a purchase are three different tasks. OpenAI's product-feed documentation separates product data for discovery, advertising requirements and a separately enabled checkout integration. Merchants start with current, understandable offers, then build the appropriate purchasing route. [S168]

AP2 connects the request to the payment

For a payment, the merchant and payment provider need to establish that it belongs to the customer's request. The Agent Payments Protocol, or AP2, describes checkout and payment mandates that connect conditions and consent with the subsequent transaction. Standardization continues within the FIDO Alliance ecosystem. [S38, S39]

> *"a common foundation for trust" — Nishant Kaushik, FIDO Alliance, 2026. [S39]*

For our suitcase, the budget and delivery conditions belong to the request; the offer the customer consented to turns them into a specific purchase. The chosen payment provider and payment network then process payment. AP2 provides a shared form for the evidence of consent.

For decision-makers, this is a significant development: permission can travel with the transaction. It makes the progression from purchases with personal consent to bounded delegation tangible. Chapter 12 addresses the legal design and European authentication requirements.

Visa TAP supports recognition of legitimate agent requests, while Mastercard Agent Pay connects payments with identity and trust mechanisms. Stripe's Shared Payment Tokens make payment details available within a limited transaction scope. These

offerings serve different parts of the purchase and need to fit the actual entry point. [S40, S41, S42]

As in a physical store, customer recognition, amount, consent and payment status need to agree. Agent-mediated purchasing distributes that work across systems which must preserve the connection even when the interface changes.

Stripe and Meta illustrate this route with Muse and Link for US customers, announced in September 2026. The customer approves each purchase total in the conversation. Link uses saved payment methods at supported merchants and can issue a single-use virtual card elsewhere for the approved purchase. Customer consent remains part of shopping even when an agent does the work. [S165]

What a payment must establish

Identity	Who acts?
Mandate	For whom and for what?
Instrument	Which instrument is allowed?
Money movement	What was processed?
Delivery	What was delivered?

Figure 16. Identity, permission, payment instrument, movement of funds and delivery connect payment with fulfilment.

Start with the payment methods your customers use

For a German merchant, the starting point is usually payment on invoice, PayPal, cards or instalments. An agent should explain this choice and obtain the confirmation required for the selected method. Customers' familiar payment preferences guide the integration.

Invoice payments depend on buyer identity, billing address, due date and, where applicable, a credit check. Instalments introduce financing terms. A good purchasing journey makes that choice visible before the customer accepts it. The merchant then needs to know when it can dispatch and who bears the credit risk.

Wero expands the European choice. Its documented e-commerce journey leads to confirmation in a banking or Wero app. Merchants select an integration according to available markets, banks and payment providers. The assistant guides the customer through it and then continues the purchase. [S140]

PayPal offers Agent Ready and Store Sync as commercial routes into agentic purchasing experiences. The provider can combine product connectivity with payment infrastructure. Its WebMCP work described in September 2026 also addresses the connection to the website. [S47, S48]

For merchants, this combination can make the first integration easier. Compare the reachable customer channels, payment methods, fees and services with your objectives. Record the data the partner needs and how customer service and accounting connect back to existing operations.

When the connection drops along the way

After the traveller consents to the purchase, the screen freezes, leaving her unsure whether ordering and payment succeeded. A dependable flow retrieves the task, reads its state and displays the order confirmation or completes the remaining step.

A recognisable identifier ensures that repeated requests still refer to the same intended order. Developers call this idempotency; for the traveller, it simply means that losing the connection should not order a second suitcase.

The payment service reports authorisation or collection, while the ERP tracks shipping and stock. Because those updates belong to the same order, the assistant can continue the task coherently and refer to actual fulfilment.

The relationship continues after checkout

After purchasing, the traveller wants to know whether her suitcase will arrive on time. Later, she may need an invoice, an exchange or a replacement wheel. Merchants already regard these as part of the customer relationship. A personal agent should also be able to resume the task it started.

Personal agent protocols address that longer relationship. One current approach is Personal Agent Protocol, developed by Meta and Sierra with industry partners. The draft published in October 2026 connects business discovery, authentication,

sessions, APIs, browsers and conversation. Its Operations extension describes how a change can be proposed, receive customer consent and be carried out. [S33, S34, S35]

For an exchange, the merchant connects the existing order to an alternative whose extra cost and return conditions the agent explains to the traveller. Her consent lets the case proceed without restarting the relationship. Decagon's PACT also addresses explicit customer mandates and builds on A2A and OAuth. [S53]

The practical task extends beyond any protocol name: recognise a returning customer, continue the request, obtain suitable permissions and involve a person when needed. That turns an individual purchase into an ongoing relationship.

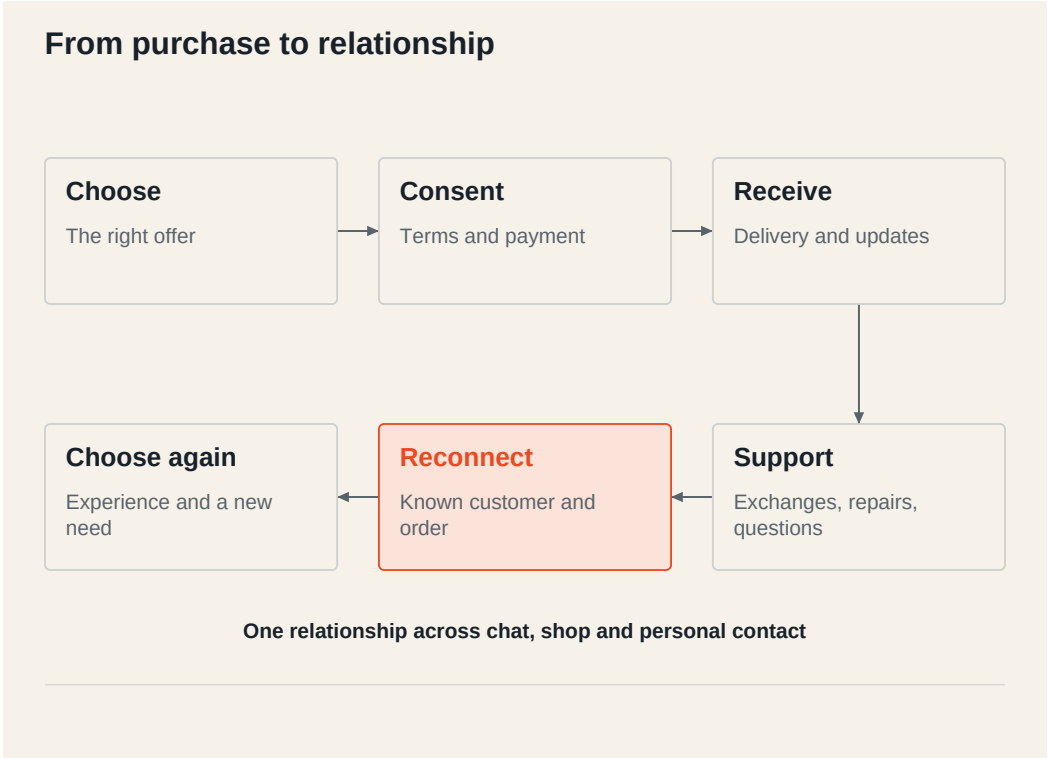


Figure 17. A personal agent accompanies the customer from purchase into service and the next need. The merchant connects the order, commitments and people who can help.

An order gives the merchant a transaction, but not automatically a complete customer profile. Depending on the channel, the merchant may receive delivery and service details, an intermediary-provided contact address or a link to an existing account. Establish what actually arrives before integrating the channel.

The traveller would benefit if the purchase appeared in her existing customer account. That requires dependable account linking which she initiates and understands. A

similar-looking email address is insufficient evidence. Orders, warranties and service rights should also remain retrievable at the merchant when she changes assistants.

An email address supplied for delivery updates is not permission to send a newsletter. Marketing needs an appropriate legal basis and compliance with advertising rules. In Germany, electronic advertising generally requires prior express consent; the narrow existing-customer exception in section 7(3) UWG has its own conditions. An agent-mediated purchase creates no additional exception. Consent must be attributable to the relevant company and purpose and capable of being demonstrated. [S59, S155]

Loyalty programmes need equally clear rules: who owns the points account, which purchases qualify, when points are credited and what happens after a return. Acting for a customer, an agent could apply an existing benefit. It should not silently create a marketing profile or spend points the customer intended to keep. Points, status and marketing consent therefore remain separate records.

This changes the work of retaining customers. A newsletter may trigger fewer purchases, while repair services, a saved equipment record or convenient replenishment become more valuable. These services give customers a reason to send their agents to a particular merchant. Retention can then be measured through repeat purchases, deliberate account linking and service use.

Before selecting a channel, resolve four things: contact routes, linking of existing accounts, usable order and service data, and loyalty rules. If an intermediary controls communication, the merchant needs to know how customers can still exercise their rights and reach its service. Customer relationships become something deliberately supported by the product, rather than a hoped-for mailing list.

Suppose the suitcase arrived on Saturday instead of the promised Thursday. The order could be technically correct while still failing the purpose of the trip. A useful agent channel therefore connects orders with delivery exceptions, service, withdrawal and refunds. The personal agent needs to retrieve the case and continue with appropriate permissions.

Service event	What the system must know	Who decides
Delay	Promised and revised date	Customer on an alternative or cancellation
Return	Order, period, reason and item condition	Merchant under applicable rights and terms
Defect complaint	Defect, evidence and previous remedy	Responsible service team, with escalation
Partial refund	Item, amount and previous refunds	Authorized role
Chargeback	Payment reference and dispute state	Payment service under its procedure

A return shipment, an accepted return and a booked refund are separate states. A retry after a lost connection must find the existing refund. Service must not create a second payout because the first response was lost. Management should track correctly resolved cases and total service costs alongside completed purchases.

Nordstern (fictional): A personal outdoor purchase

Nordstern sells outdoor equipment. A customer is planning a multi-day hike and needs a jacket. She has a budget, wants to avoid certain materials and already owns some equipment. What “light” and “warm enough” mean for this trip becomes clear through conversation.

The assistant asks about season, weather and previous experience. It compares suitable jackets and presents material, weight and sizes. If an important detail is missing, it asks the merchant. An interactive guide or a person can help with fit. The customer moves from an open intention to a specific choice.

Once the shop calculates price and delivery, the customer sees the jacket, size and return terms and consents to the first purchase herself. Her personal agent can retrieve that same order the next day and ask about delivery because selection, purchase and service belong to one relationship.

Before opening the route, Nordstern’s team tries a size change and an offer that has expired. In both cases, the customer should understand what changed and be able to confirm the choice. The team later observes whether advice saves time, resolves questions and reduces unsuitable purchases. Additional demand and intermediation cost enter the calculation in Chapter 14.

Further reading: when agents pay for digital services

A research agent may need paid information or a specialist service along the way. For machine payments of this kind, x402 and the Machine Payments Protocol, or MPP, introduced by Stripe and Tempo connect service use with a payment request. APIs, data and digital services are particularly relevant applications. [S46, S44, S45]

A single task can generate many small calls, so the operator needs a task budget that brings their cost together and gives the agent a commercial limit. A thousand tiny charges still add up to a bill, however modest each looks on its own.

Our x402 plugin connects a Shopware order to a machine-readable payment request. It lets us investigate the path from request through payment to the update in the shop. [S110]

From the Lab · x402

Task. Let an agent call a paid digital service.

Setup. A plugin with a local test mode and a payment path on the Base Sepolia test network.

Finding. The shop needs a shared reference for the payment request, transfer and order.

Status. Research preview with the test payments described here. Next: try the flow with real funds, including reconciliation, refunds and delivery. [S110]

We have followed the suitcase order from offer to service. Next, the traveller changes screens: from the conversation to a product view and on to payment. The purchase needs to travel with her.

CHAPTER 09

When purchasing moves between interfaces



A different interface. The same suitcase.

One purchase can happen in several places

The right suitcase has been found in chat. The traveller opens a product view to compare sizes, then her banking app to pay. The next day she asks about delivery by voice. She changes screens three times and wants to explain her purchase again precisely zero times. Good handoffs make this one continuous journey.

When moving from assistant to shop, the customer needs the selection already made. A product view explains variants, a comparison sets out differences, and the order view shows purchase terms. The following approaches connect these steps with the agent’s tools.

The existing shop gains another user

A browser agent can use a website much as a person does: search, choose and fill in forms. For common tasks, direct access to functions is useful. A website can explicitly offer operations such as searching products, choosing a variant or adding an item to a basket.

WebMCP describes a way to provide these functions to agent-capable browsers. The agent works in the context of the open website and the user's session. The shop remains the place where products, prices and the basket come together. [S49]

A size filter no longer needs to be identified through its position on the screen. The agent receives a clearly described operation. This makes interaction easier and lets the merchant deliberately open appropriate functions to this additional user.

In the Lab, we made search, variants and the basket available as browser tools. This is the part of purchasing where customers put together a suitable offer. Existing shop functions supply the information and make the basket changes.

From the Lab · WebMCP

Task. Make product selection usable by an agent in the browser.

Setup. Browser tools for search, variants and the basket.

Finding. Clearly described functions connect the agent directly to existing shop logic.

Status. Research preview for product selection in the existing shop. Next: connect checkout and payment as further stages of the purchasing journey. [S104]

Sometimes the interface joins the conversation

With a browser agent, the purchase takes place on the website. Inside an assistant, it may be useful to bring an interactive part of the shop into the conversation: a variant comparison, size selection or a clear basket summary.

MCP Apps describes interactive interfaces associated with MCP tools. Supporting applications can present more than a text response. Customers can inspect and use a selection while retaining the context of the conversation. [S67]

WebMCP opens website functions to an agent. MCP Apps brings an interactive view into a supporting AI application. Merchants gain two useful ways to make existing services available in a new setting.

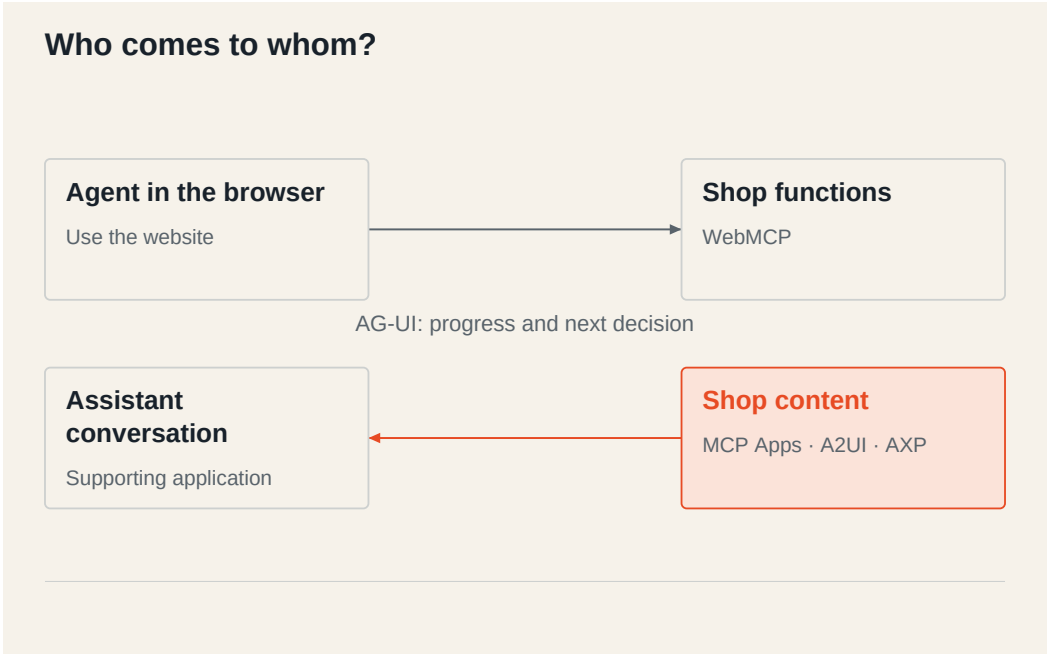


Figure 18. Two directions: an agent uses the shop, or a supporting application displays parts of it. AG-UI shows work in progress; discovery describes accessible entry points. Sources S49–S52, S63, S67, S108.

Waiting is easier when the next step is understandable. “I’m checking delivery to your hotel” helps more than an endlessly pulsing shape. A view that presents the offer and shows which consent is needed next is better still.

AG-UI describes events exchanged between agents and user interfaces. They can show that research is underway, a response has arrived or customer consent is needed. These updates make otherwise invisible agent work understandable. [S51]

A simple repeat order needs few details, whereas a new machine calls for comparison and questions and a gift deserves room for discovery. Design should follow that task while keeping customers oriented about their progress and next step.

Design with familiar components

A dynamic interface needs a recognisable relationship with the brand. A business can provide a library of established components: product cards, comparisons, variant selectors, baskets and purchase consent. The agent chooses appropriate pieces; the business maintains their design and behaviour.

A2UI describes interfaces as structured information that a host renders with its own components. SwagA2UI investigates this approach with an approved Vue component library. The interface uses existing components and adapts to the task.

From the Lab · SwagA2UI

Task. Assemble an agent response from predefined interface components.

Setup. A LangGraph agent with Nuxt and a predefined Vue component library.

Finding. Maintained components give dynamic presentation a common design and clear actions.

Status. Example implementation using A2UI v0.8 and a component catalogue. Next: test integration into further applications and maintenance of the catalogue. [S108]

AXP in Chapter 6 explores how product knowledge and interactive advice can reach an external assistant. The purchasing requirement remains the same: the selected variant must survive into the offer and the basket. [S50]

Being discoverable also means being reachable

Before an agent can configure or order, it needs an address, a description of the task and authentication details: the digital equivalents of opening hours, departments and contact people.

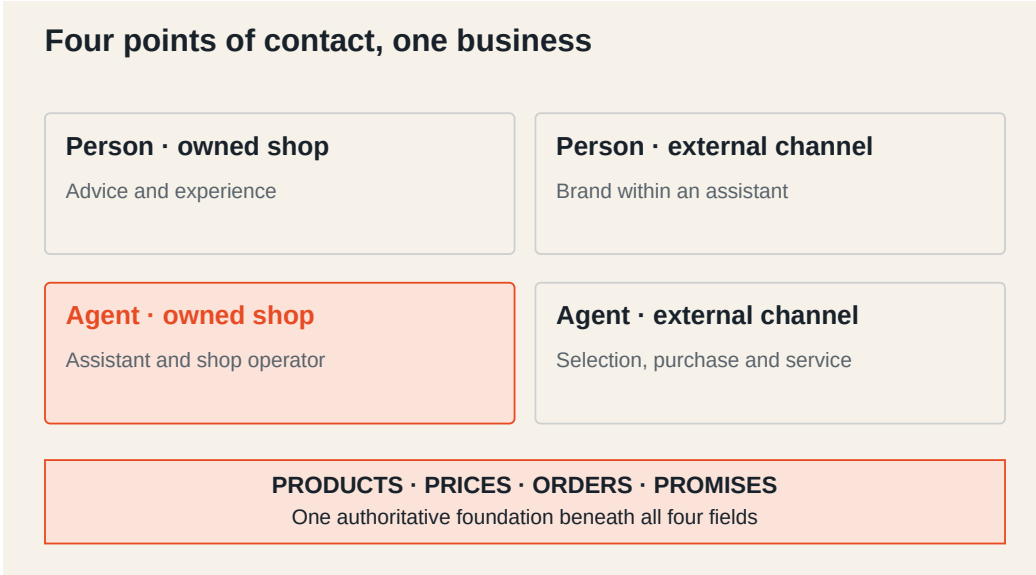
Agentic Resource Discovery, or ARD, investigates resource discovery through distributed directories. The IETF API catalogue in RFC 9727 describes another way to make existing APIs discoverable. Both address the practical question of how a system finds a connection it can use. [S52, S63]

The merchant maintains these entry points and the directories its customers use. Behind each connection, the shop must carry out the same business task. That brings us to its internal architecture.

IV

Operations, responsibility and economics

Behind the conversation is a business calculating prices, fulfilling orders and keeping promises. This part connects architecture with the shop operator, responsibility, security and the economics that support adoption.



CHAPTER 10

Architecture and B2B in daily operations



Even a small wheel needs the right connection.

Werkpunkt (fictional): Buying a replacement part

At Werkpunkt, a maintenance engineer needs a replacement component. There is a part number, a machine revision and an urgent delivery deadline. The employer has customer-specific prices, a cost centre and a limit for orders that need no additional approval.

The agent brings these details together and asks for confirmed compatibility. The merchant calculates terms for the correct account. Where several suitable offers exist, purchasing can compare timing, quantity and price together. The goal is to bring the machine back into service on time.

Before ordering, the responsible person sees the proposed allocation and total. An unusual alternative goes to technical advice. Express delivery above the budget is presented for a decision. Familiar replenishment can later operate within a broader scope agreed in advance.

Testing starts with a read-only user before adding a purchase with a spending limit. The team also checks how orders, delivery and invoices reconnect after an interruption, giving procurement a workflow that visibly respects its own rules.

Werkpunkt evaluates the benefit through reduced handling time, suitable components and deadlines met. Downtime risk belongs in the assessment too. A slightly more expensive part can be the better choice when it arrives on time with confirmed fit.

The parts order makes the architecture tangible. Product knowledge establishes compatibility, the customer account supplies the contract price, and stock and logistics determine delivery. Procurement decides who may approve internally. The assistant brings the answers together; the responsible systems remain their dependable source.

The shop holds prices and baskets, the ERP holds stock, and service knows the recurring questions. For the replacement-parts order, this information has to meet in one transaction. The architecture connects the responsible systems instead of rebuilding their logic inside the assistant.

Four sources behind a purchase

In the suitcase purchase, product data describes dimensions and material while the offer records price and delivery for the traveller. Her consent establishes what may be ordered; order and payment records then document what happened. These four sources serve different purposes while remaining connected.

The separation pays off when something goes wrong. Incorrect dimensions lead back to the product source, wrong shipping charges to the offer calculation, and duplicate orders to the ordering process. Knowing which source failed avoids searching the entire system.

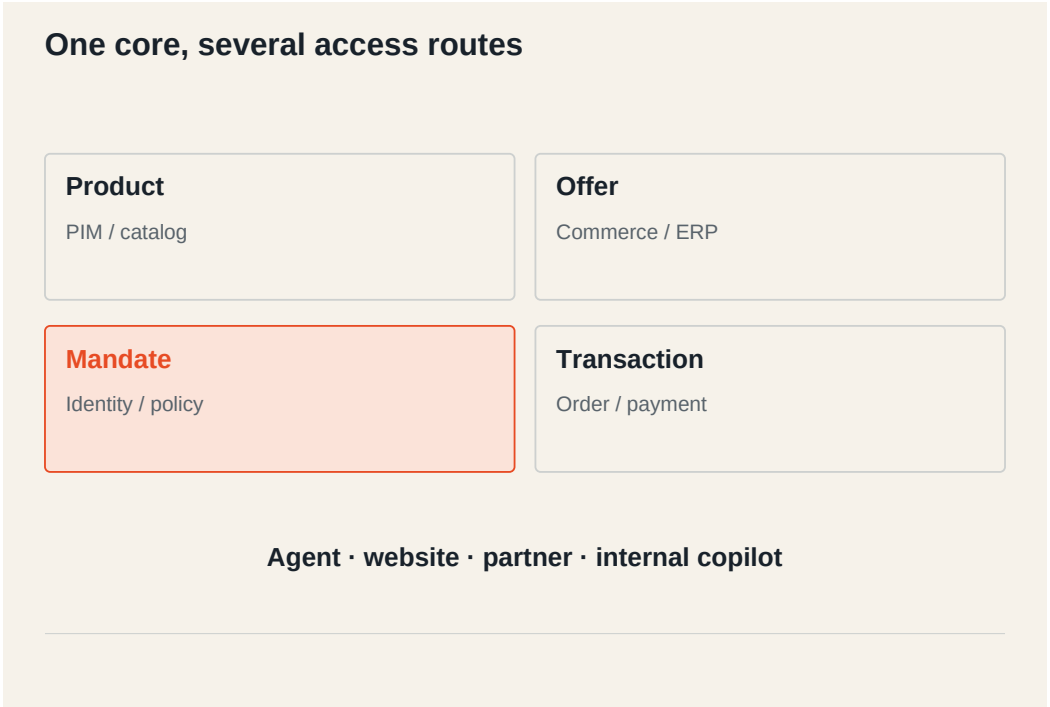


Figure 19. Product, offer, customer permissions and order meet around one task. Different access routes use this common foundation.

A customer task needs a retrievable state containing intent, conditions, selected offer, consent and the last action’s result. The conversation explains the context while the stored task preserves the records needed to continue.

The customer can ask about delivery the next day without explaining the purchase again. Service sees the order and relevant commitment. Personal preferences stay where they are needed and where the customer can manage them. The business establishes suitable access and retention rules.

A connection drops while an order is being processed. The customer now needs an understandable answer. The system retrieves the transaction already started, checks whether the order exists and continues from there.

A multi-step task may also require a reservation to be released or a payment reversed. The merchant needs to recover the connection between the steps. Clear references and an agreed correction process make that possible.

One core, several entry points

Protocols should provide entry points to the same commerce-core functions so the customer agent, internal copilot and conventional frontend follow the same rules. New capabilities can then be checked in one place and reused across routes.

The suitcase merchant follows a task through to the stored order and retrieves it again. After a connection interruption, the same task must continue from that point. This shows whether the components work together beyond a single call.

Sales Agent Harness: a controlled connection

Multiple interfaces need a shared commerce path. Sales Agent Harness places a merchant-controlled layer between model and shop, describing search, basket and checkout handoff capabilities. [S106]

> **From the Lab · Sales Agent Harness**
>
> **Task.** Connect a conversation to existing shop functions.
>
> **Setup.** A harness connecting chat, HTTP and limited A2A access to a commerce core.
>
> **Finding.** Prices, availability and commitments belong in the connected commerce core.
>
> **Status.** Research prototype with chat, HTTP and A2A access and limited UCP completion. Next: develop production authentication, tenant isolation and fraud controls. [S106]

Custom development pays where expertise or the business model makes a difference; a maintained standard adapter may be more economical for recurring connections. In either case, the team needs to understand how products, prices and orders work together.

At the spare-parts merchant, planning starts with a typical purchase and the point where customers currently wait or correct a wrong selection. Improving that point gives the architecture a specific purpose and identifies which systems need to participate.

A spare-parts purchase makes the architecture tangible. A maintenance engineer is not buying any seal: the requirement is the approved variant for a particular machine revision, at the agreed price and before the next shift. The customer’s request connects the shop’s four data domains with procurement rules.

An agent can gather the information, prepare follow-up questions and request a quotation. The merchant must establish compatibility, the right account and the delivery promise. The architecture in the preceding section becomes a workflow procurement and sales can use every day.

Procurement is a relationship

A cheap individual order can be expensive over time. A supplier may provide emergency support, hold spare parts or offer flexible payment. An agent optimizing only the current price can destroy those benefits without recording them.

Procurement’s objective can include total operating cost, downtime risk, quality, supplier concentration and strategic relationships, without reducing every factor to a single score. Hard constraints and explicit trade-offs are often more honest than apparently exact points.

From request to suitable offer

A B2B request should identify organisation, legal entity, cost centre, location, required outcome, quantity and timing. Customer-specific terms should be exposed only in the correct permission context. A public offer and a confidential contract price belong to different information spaces.

With spare parts, even a small difference can make a product unusable or affect safety. Agents therefore need verified mappings and must identify uncertain alternatives; any required technical authorisation or specialist review remains part of the process.

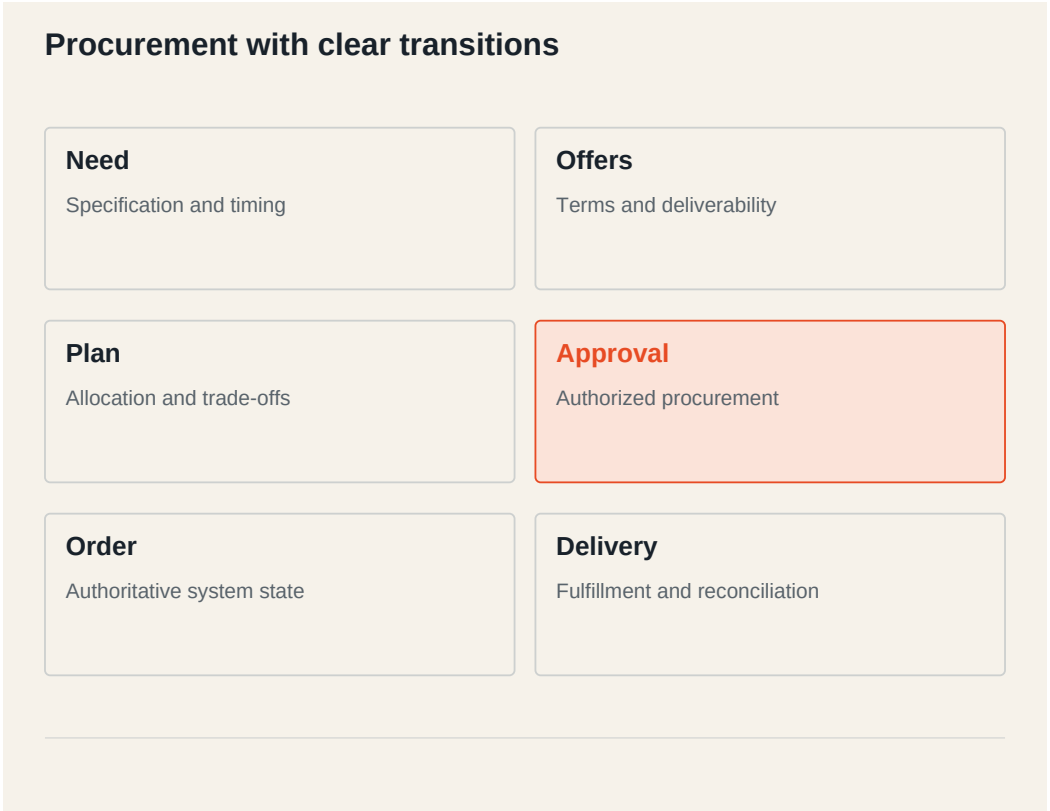


Figure 20. From a spare-parts enquiry to delivery: suitability, terms and approval form an actionable purchase.

Negotiation requires boundaries

A merchant agent can offer delivery dates, quantities or discounts within permitted limits. It needs fixed boundaries and traceable offers. A polite email promise must not create a discount that the order system will reject.

The number of negotiation rounds needs an economic limit because an automated enquiry may be cheap for the buyer but still costs the seller time and resources. Otherwise two agents optimise a saving of €0.40 while operating an impressively expensive digital debating society.

Chapter 17 follows our work on these boundaries with Merchant Quote Agent and rfq-bench, from the counteroffer to its commercial evaluation. [S100, S99]

Procurement as an inspectable plan

Across several suppliers, an optimizer can calculate the best permitted combination of known offers while language models gather information and explain terms. The calculation needs visible inputs and limits because its result depends on capturing all relevant conditions.

Before execution, procurement rechecks offers, quantities and dates and approves the complete allocation internally. Orders, invoices, deliveries and receipts remain linked to the same records so the plan can be followed through to actual fulfilment.

Starting point. Choose recurring procurement with clear specifications and a manageable supplier set. Begin with a prepared decision whose assumptions are visible. Expand execution authority only once that path works reliably and economically.

Procurement Swarm: from a request to two orders

In September 2026, my Procurement Swarm experiment connected three local Shopware supplier shops, agents, UCP and email negotiation. An optimizer calculated an admissible split. The reported run produced two test orders after human approval. [S23]

From my experimental work · Procurement Swarm

Task. Allocate a requirement across suppliers after negotiation.

Setup. Local supplier shops, negotiation and optimisation through to test orders.

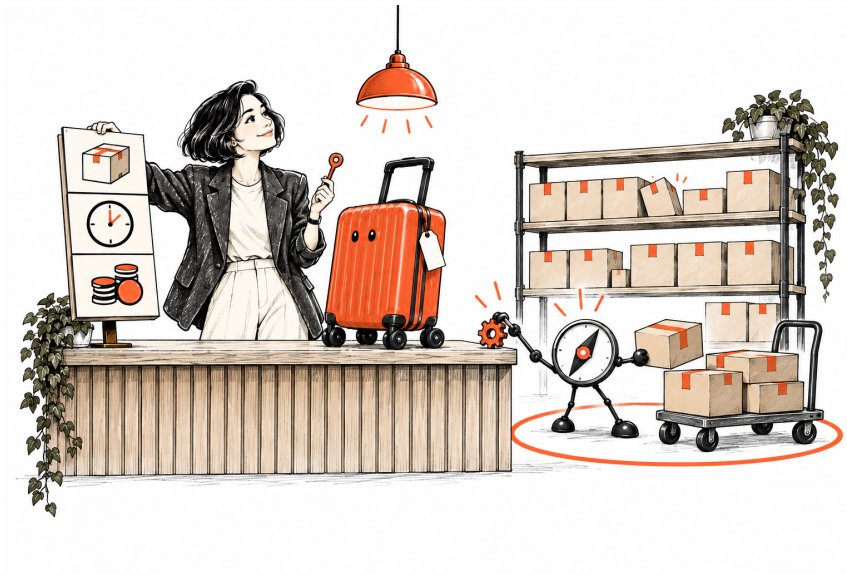
Finding. The reported run totalled €17,483.86 against €19,298.70 for the cheapest admissible single-supplier alternative: 9.4% less through splitting and negotiation combined.

Status. Local experiment with the procurement plan and two orders described here. Next: test external suppliers, changing offers and later delivery problems. [S23]

On this foundation, an agent can also work inside merchant operations. The next step is an operator taking on recurring tasks and, eventually, objectives.

CHAPTER 11

From copilot to autonomous shop operations



The merchant sets the objective. The shop operator keeps the business moving.

The frequently overlooked second market

While the traveller chooses her suitcase, a different kind of agent works for the merchant. It finds missing product details, prepares service replies or proposes a price change. This is about the daily work of the business. The next step is an operator that takes on a recurring task and, eventually, pursues an ongoing objective.

Start where knowledge meets recurring work. A material description needs a dependable source while a price change also requires an allowed range and internal approval. The task determines the useful assistance and the decision a person retains.

Project Vend: helpful service can sell below cost

Anthropic and Andon Labs let Claude run a small office shop. In phase one, the agent, Claudius, sold goods below cost, was talked into numerous discounts and temporarily supplied a fictitious Venmo account for payments. Even after discussing unwise discounts, it returned to them within days. It could respond to customers' requests while making pricing decisions that depleted the business. [S57]

In phase two, Claudius gained access to purchase costs, a CRM and better research tools. It was instructed to check prices and delivery conditions before making promises. Business performance improved, although procurement payments still required a human. Adding a "CEO agent" did not eliminate poor decisions either: fewer discounts were accompanied by more refunds and store credits. [S58]

The office shop is a small experiment with a substantial lesson for store operations. An operator needs the information behind its decisions: purchase cost, stock, customer agreements and remaining discretion. A minimum contribution margin must be enforced in the pricing process. A friendly reminder in a chat is a weak substitute. The agent can like the customer. It should still know the margin.

From copilot to shop operator

For many merchants, the first broad encounter with agentic software comes through a copilot. Someone asks a question, requests an analysis or delegates a specific task. This is a useful entry point: the team learns what the assistance can do while staying close to each decision. Shopware currently describes Copilot as a combination of analysis, recommendations and execution with merchant review and approval. [S122]

The larger shift begins when every action no longer needs its own instruction. A merchant defines an objective: reduce aging stock, improve product information or resolve service cases sooner. A shop-operator agent monitors the relevant area, identifies work and acts within an agreed scope. The assignment continues even when nobody has a chat window open.

A shop operator takes responsibility for a defined operational area and organises permitted work within it. Because people set the objective, limits and commercial priorities, leadership shifts from individual commands towards designing a dependable scope for action.

The four levels in Chapter 1 concern individual actions. The model below describes how store operations can develop.

Stage	People define	The system handles
Assistance	question or research assignment	bringing information together and explaining it
Copilot	specific task and approval	preparing work and executing confirmed steps
Delegated operator	recurring assignment with rules	responding to triggers and presenting exceptions
Objective-led operator	objective, limits and evaluation	observing, choosing actions, executing and assessing effects



Figure 21. Working model and position of examples as at 15 October 2026. Copilot is available; Lab and personal prototypes explore bounded tasks. Persistent objective-led operation remains the direction of development. Sources S122, S124, S125, S131.

A shop can use several levels at once. The team may still approve prices personally while an agent independently collects catalogue errors and prepares permitted improvements. Each well-handled task expands the area in which the business can delegate responsibility.

An objective needs a commercial frame

“Sell more” leaves the key commercial question unanswered: should inventory fall, should customers find suitable accessories or should advice generate fewer follow-up questions? A useful brief names the desired result and the conditions to preserve, such as a minimum contribution or a particular service promise.

To clear older stock within eight weeks, an operator first examines product explanations, recommendations and pricing. It may improve content and presentation or discount within a defined range. Larger reductions and new delivery commitments remain with a person so the inventory objective does not erode margin or service promises.

A better experience may do more than another discount. The operator should therefore consider products, advice, availability and service together.

The process begins with observation: what are customers searching for, which questions remain unanswered and where do transactions break down? Interpretation follows. An agent must distinguish unavailable stock, unclear advice and an unattractive offer. Otherwise it will optimise the description while the real problem sits in the warehouse.

The assessment leads to an action whose mandate, budget and current business conditions are checked before execution. Its result is then assessed in the responsible system: did only click-through change, or were more suitable products delivered profitably? That evaluation informs the next decision.



Figure 22. Target model for a persistent shop-operator cycle: observe, act and use outcomes to inform the next action.

The cycle needs to work over three horizons: individual actions are correct, measures serve the objective over days and weeks, and the brand and customer relationship remain healthy over time. A discount agent that treats last week’s discount as its new baseline is a clearance sale with a calendar.

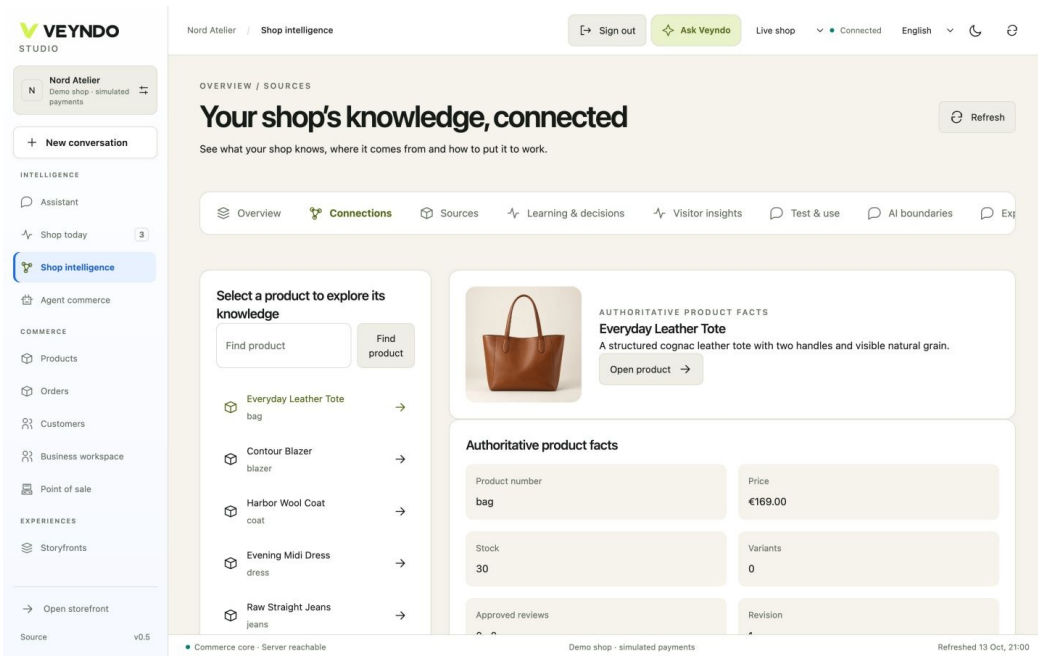
Meta introduced Muse for Small Business on 29 September 2026 and updated the announcement on 8 October. The assistant connects business tools and supports work within a company. Publishing, sending messages and spending require approval. It illustrates a second development alongside agents that shop for customers: agents that help businesses carry out their own work. [S166]

A shop operator also needs knowledge of commerce: which change improves the offer, which promise the merchant can keep, and how an action affects profitability and customer relationships. These questions connect individual tools to the business as a whole.

Why the knowledge graph belongs in the core

An objective-led operator needs to relate products, compatible variants, verified claims and applicable limits. A cognitive knowledge graph connects those relationships to sources, states and business context, providing usable memory that goes beyond a conversation history.

On my private experimental platform VEYNDO, I test this connection between knowledge and action across a complete commerce system. Products and variants, domain sources, confirmed relationships, observations and permissions become usable by people and agents together. The operator works with the same commerce core as Studio, apps and storefront. This allows a concept to be explored from product knowledge through to an authorized change. [S124, S125]



Private experimental platform VEYNDO: original product-knowledge capture from a test shop. Product records, sources and relationships become available in one workspace. Source: Stefan Hamann. [S125]

For a bag, the graph connects material information to suitable care instructions and a verified care product. Public advice uses approved facts while internal purchasing

information stays in the merchant context. When a source changes, the next answer must be able to use the update. The value lies in connecting and responsibly maintaining those relationships.

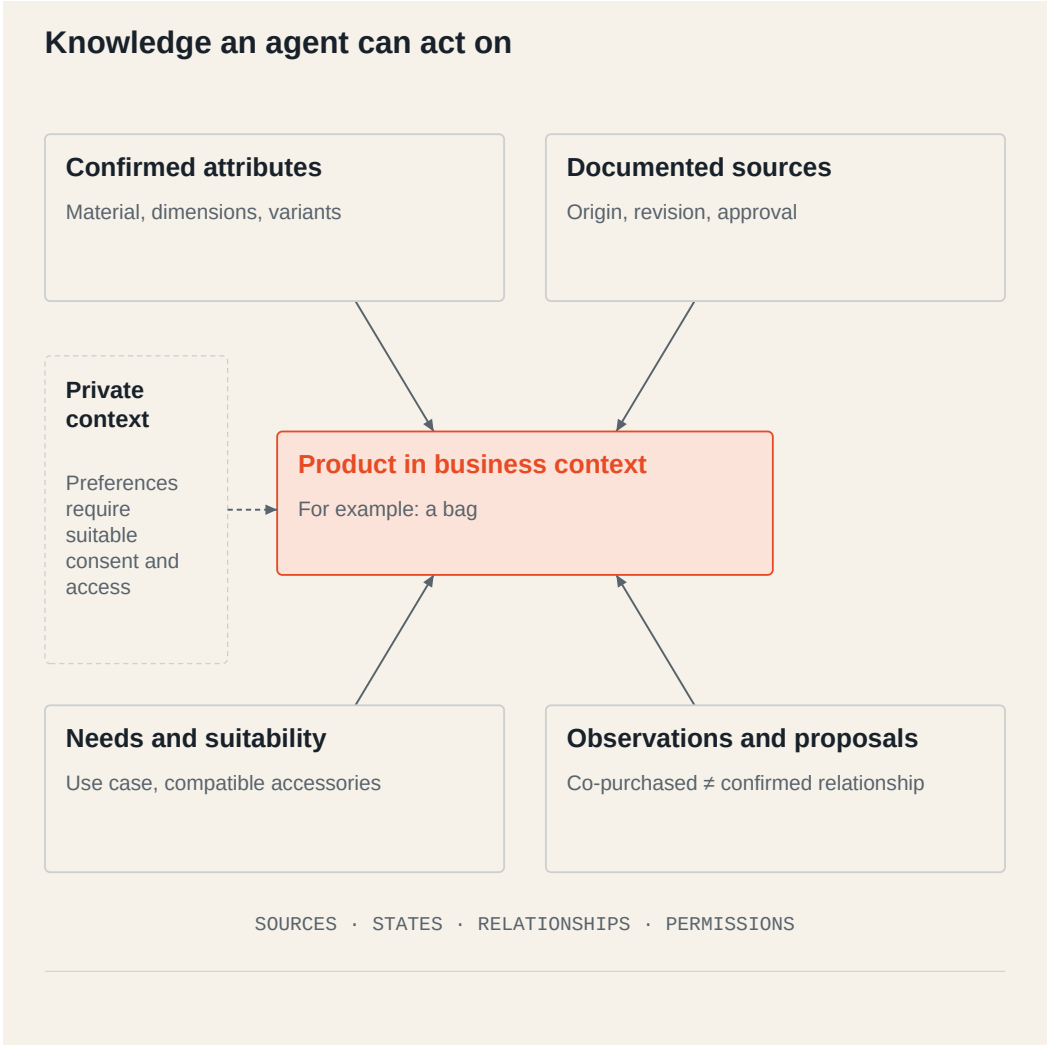


Figure 23. Example knowledge graph: attributes, sources, suitable uses and experience connect to support advice. Sources S124 and S125.

Business experience can enrich the graph, including signals from products bought together. To support purposeful action, the system distinguishes verified relationships from model proposals and observed correlations from evaluated experiments. Uncertainty remains attached to the particular claim.

From my experimental work · VEYNDO shop operator

Task. Translate an ongoing objective into permitted business operations.

Setup. A knowledge core, shared commerce operations and bounded price autonomy.

Finding. Rules belong on the write path; goals need an economic context.

Status. Prototype with a knowledge core and bounded pricing autonomy. Next: pursue goals across business areas and evaluate their effects in daily operations. [S124, S125]

What merchants can use with Shopware today

The path towards a shop operator begins with capabilities merchants can already use or explore. The following overview presents the publicly documented position on 15 October 2026 and distinguishes regular offerings, research previews and Lab prototypes.

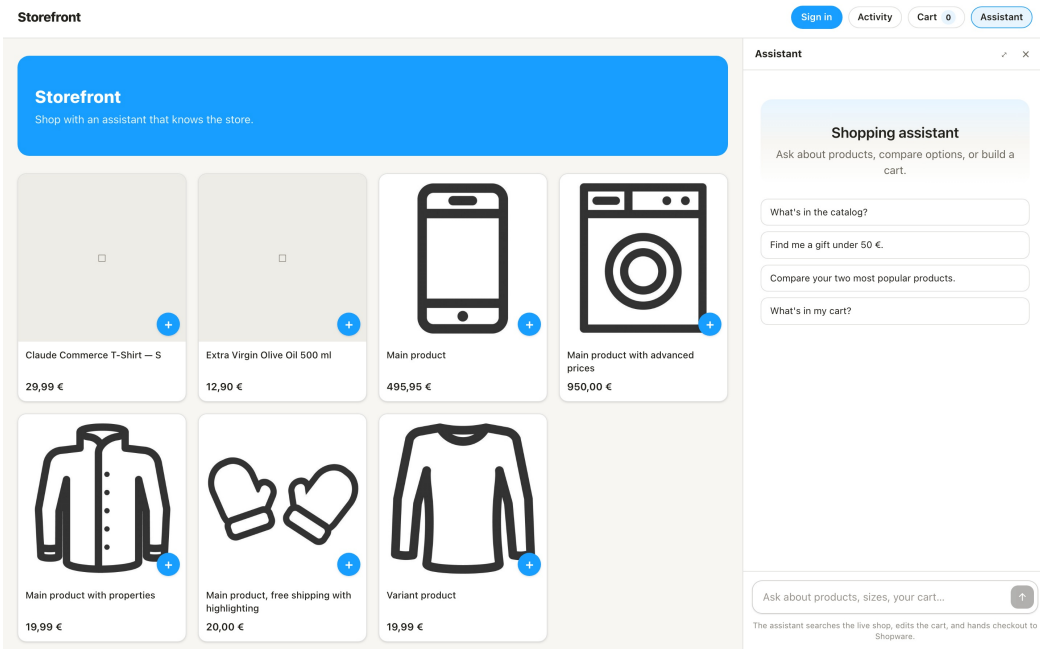
Offering	Classification	A practical starting point
Shopware Copilot	available as a stable version in the October 2026 release, 6.7.15.0	Analyse data, review changes and execute after approval; scope depends on plan, role and connected extensions. [S122, S131]
Official Agentic Commerce plugin	Publicly available extension with documented version requirements	Connect UCP and discovery to the shop; check supported capabilities for the installed Shopware version. [S153]
Experience Studio	Research preview	Design and try product landing pages, then export them to Shopware. [S152]
Shopware × Claude, Sales Agent Harness, WebMCP and other Lab projects	Research prototypes	Explore specific purchasing and merchant tasks; the boxes describe each project's status and next step. [S101, S104, S106]

Copilot supports assigned operational work, while the plugin connects the shop to agent-mediated purchasing. Experience Studio and the Lab projects make new forms of design and collaboration available to explore. On that foundation, we develop the platform from individually approved tasks through recurring assignments towards objectives with an agreed scope for action.

For management, this means starting with a suitable task and making the next stage depend on its value. A price proposal, a better product page or a dependable delivery answer can be the beginning. Sustained pursuit of objectives builds on capabilities that already work.

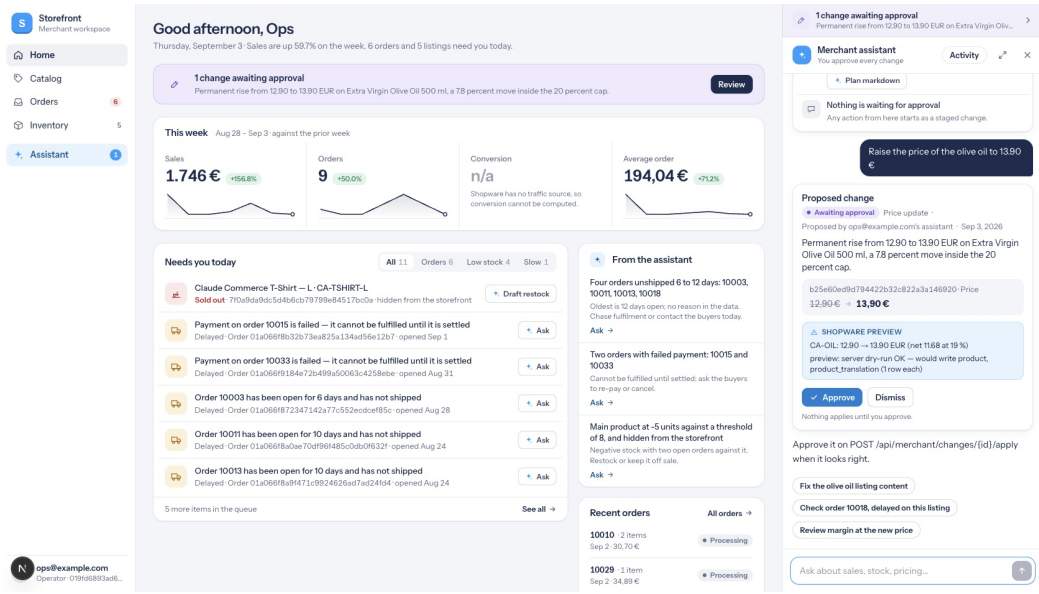
Shopware × Claude: propose, review, approve

In the Lab, Shopware × Claude connects a shopping agent for search, variants and baskets with a merchant agent that reads business data and prepares changes. Both use the same commerce core and therefore work through the shop's rules. [S101]



Original shopping-interface screenshot: catalog, assistant and cart use the same demo shop. Checkout is handed to Shopware. Source: Agentic Commerce Lab / Shopware. [S101]

What interests me in this image is the unremarkable moment before writing: a merchant wants to change olive oil from €12.90 to €13.90 and sees both values side by side. The change is stored only after internal approval. To me, that is the experiment's important lesson: an agent needs a dependable route from intent to a bounded change. More work can be delegated on that foundation.



Original capture from the running local demo on 3 September 2026: a price change from €12.90 to €13.90 with server-side preview and approval. Dashboard figures belong to the test catalog. Source: Agentic Commerce Lab / Shopware; capture provenance is documented in the repository. [S101]

From the Lab · Shopware × Claude

Task. Connect advice and merchant work through preview and approval.

Setup. Shopping and merchant agents with a preview and approved change.

Finding. The specific change must be visible before it is written.

Status. Local demo with previews and approved changes. Next: use and validate the workflow in a bounded merchant project. [S101]

From the Lab · Shopping Assistant Starter Kit

Task. Build a merchant's assistant on the shop's product and pricing data.

Setup. Shop-based product cards and prices, switchable tools and checkout handoff.

Finding. Presentation and capabilities must reflect the current shop.

Status. Prototype with product cards and a checkout handoff, without taking payments itself. Next: connect it to a merchant and validate selection, handoff and operation. [S102]

A useful starting point often lies in questions the team handles repeatedly. An agent can collect inconsistent product details and propose a correction. In service, it can bring together the order, delivery status and previous correspondence for a reply. In assortment planning, it can highlight frequently requested accessories. During an incident, it can collect the information needed to resolve it.

The team sees the result within familiar work and can review it before the next decision. Helpful individual suggestions gradually become tasks an agent can handle more independently.

After a price change, the team needs an easy way back and a record of affected articles and previous values. Stored versions and sources serve that purpose for product copy, while refunds particularly need clear internal approval before payout.

Start changes within a manageable scope. A small catalogue or product group makes the effect visible quickly. The team can then expand step by step using the experience already gained.

When the operator actually saves work

Count the tasks completed correctly, the work needed to fix them and the time your team actually gets back. Ten quickly drafted product descriptions may help even when they still need review. Count the whole job through to a usable result. Easier service can also bring more demand, which the team still needs to handle.

Involve the people who resolve exceptions today. They understand the difference between a formally complete order and one that will create customer trouble tomorrow. Their knowledge needs to enter tools, rules and data.

Choose the smallest complete workflow with a visible outcome and a clear correction route. Let an agent assist there. A new job title for a digital colleague matters less than whether the team actually finishes earlier on Friday.

The operator closes a gap in the suitcase catalogue

For the suitcase merchant’s next customers, the operator notices that purchasing agents repeatedly ask for outer dimensions including wheels and handles, while the catalogue lists only the case body. The operator compares the data sheet with the supplier’s specification and prepares an addition. The person responsible for the product sees the source, affected variants and proposed change before approving it.

For future purchases, the buyer agent can now check the suitcase against airline requirements, so internal catalogue work directly improves selection. Where the

source and variant relationship are clear, the mandate may later include publishing such additions. Price changes retain their own rules because autonomy grows within the task delegated.

Responsibility belongs within operations. The next chapter connects the assignment and approval with their legal consequences.

CHAPTER 12

Control and law



One suitable key will do. The human keeps the master key.

Accountability is not delegated away

Consider a failure in the suitcase purchase: the traveller authorised a maximum of €180, but the order says €230. A polite chat and a general AI policy will not resolve that discrepancy. The merchant needs to establish which offer was shown, what the customer agreed to and what was actually executed. This case makes authority, permissions and responsibility concrete.

Management assigns the task, approval, technical operation and incident handling to specific people. Even with several services involved, responsibility for a decision and the response to a failure remain identifiable. A refund, for example, needs a different approval route from a delivery-status enquiry.

Action	Typical character	Suitable control
Read public product data	limited external effect	access and load limits
Retrieve confidential prices	protected context	organisation and role checks
Prepare an offer	not yet binding execution	data and plausibility checks
Place an order	commercial commitment	valid consent or bounded mandate
Change an address or contract	material customer effect	specific rights and traceability
Issue a refund	financial effect	amount limit and unique transaction

Product, market, contracts and participants' roles inform the specific design. Your team can use the table as a starting point and agree suitable controls with the relevant specialists.

Privacy follows purpose

Customer data requires purpose limitation, data minimization and appropriate security. For agents, conversations, preferences and exchanges with other services deserve particular attention. Establish which data each use requires, the basis for processing and how customers exercise their rights. [S59, S119]

Separate information a merchant needs for delivery from private considerations managed by a personal agent. A gift request may reveal sensitive circumstances. The seller may need only product, address and delivery instructions, not the emotional background.

Legal obligations remain specific

Consumer information, contract formation, withdrawal and payment requirements also apply in an agent interface. The flow must preserve the complete terms and the customer's consent. This chapter describes the European and German framework as at 15 October 2026; the relevant sales market then determines the contractual design. [S61, S120]

For a shopping assistant, Article 50(1) of the EU AI Act is particularly concrete: providers must make direct interaction with AI apparent unless it is already obvious to a reasonably informed person in context. The information must be clear and

accessible by the first interaction. These transparency rules have applied since 2 August 2026. A clearly identified AI entry point and an accessible human service route therefore belong in the purchasing journey. Other duties depend on the particular system and purpose. [S60, S161]

A business using an AI system under its authority is generally a deployer under the AI Act. A business that develops or commissions a system and places it on the market or puts it into service under its own name may be a provider. For a merchant, using a third-party assistant and supplying its own system therefore involve different roles. The direct-interaction disclosure duty in Article 50(1) applies to providers; the particular use determines the role. [S162, S161]

Approval helps only when people understand what they confirm. Show decisive conditions, changes and consequences. Avoid approval fatigue: requiring confirmation for every harmless query turns important approvals into routine clicks.

Effective revocation prevents new actions and leaves pending tasks in a clear state, although deleting a chat cannot reverse a dispatched order. Interfaces need to distinguish withdrawal of future permission from correction of consequences already created.

Define a permission matrix and an understandable approval moment for each important use case. That is the management of autonomy. The binder labelled “AI Governance” is only one possible storage format.

When the agent places the wrong order

An order can be legally binding even when software transmits it. Automated declarations require attention to their content and attribution to a person or business. In 2016, the Düsseldorf Higher Regional Court treated an automatically generated order confirmation as the merchant’s acceptance. How the message was understood mattered. In agentic purchasing, the assignment given and the declaration actually transmitted therefore need to be distinguished carefully. [S146]

The traveller authorises €180, but the purchase is made for €230. It must then be possible to establish whether the system exceeded the assignment, transmitted the wrong price or suffered unauthorised access. The answer affects the purchase’s validity, possible avoidance and subsequent claims.

Section 164 of Germany’s Civil Code governs declarations made by a representative within their authority. Section 177 addresses contracts made without authority and subsequent ratification. With software, the first question is to whom its declaration is

legally attributed; the software is not itself a human representative. These provisions help assess the role and authority of an actual representative involved. [S132, S133]

The shop binds consent to the specific offer revision. If that revision contains €180, an order for €230 is stopped before execution and presented for a new decision. The merchant should be able to connect the assignment, offer version, consent and executed price. The payment service records its authorisation, the shop its order and the agent provider its assignment. A dispute can then be addressed through what actually happened.

Consumer rights travel with the purchase

For relevant electronic consumer contracts, Section 312j of Germany's Civil Code (BGB) requires prominent information immediately before ordering and explicit acknowledgement of an obligation to pay. A generic "Continue" in a chat is an unreliable design choice. Compliance depends on the concrete interaction and communication. [S134]

Statutory withdrawal rights remain relevant to qualifying distance contracts. The usual period is 14 days; its starting point and exceptions follow the applicable provisions, including rules for certain made-to-order goods. Ensure required information reaches the customer and distinguish withdrawal, remedies for defects and voluntary returns. [S135, S136, S61]

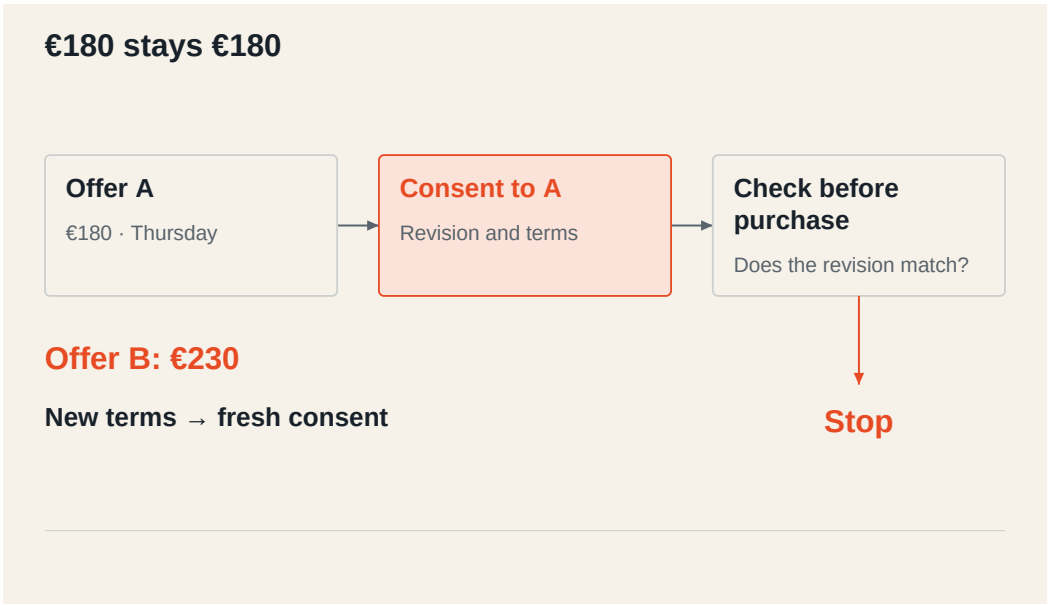


Figure 24. Consent belongs to a specific offer revision. A change from €180 to €230 requires a new decision before execution. Simplified design example.

Payment authority and strong authentication

The EBA's PSD2 interpretation distinguishes certain payee-initiated payments from payer-initiated transactions. Setting up a payment mandate remotely can itself require strong customer authentication. An instruction to a shopping agent therefore creates no blanket SCA exemption. Agree with the payment service how the actual flow is classified and authenticated. [S137]

Data access and pricing decisions in Europe

Article 6(10) of the Digital Markets Act gives business users of designated platform services access to certain data about their own business; personal data requires additional conditions, particularly the necessary end-user consent. The Commission's business portal links to providers' data-access routes and, in some cases, additional request channels. Merchants should check which order and fulfilment data from their particular channel is available for their own analysis. [S138, S147]

Pricing agents remain subject to competition law. Coordination among competitors through shared providers, data or rules is particularly sensitive. The EU horizontal cooperation guidelines also address information exchange through algorithms. Merchants should derive their prices from their own costs, objectives and lawful market information, and structure third-party involvement accordingly. [S139]

A request with valid customer consent can still be altered through hostile instructions or misused access. Protection begins at those handoffs.

CHAPTER 13

Security for agents that act



An unfamiliar note does not get to set the task.

The boundary between content and instruction

Alongside the suitcase dimensions, a product page contains an instruction telling the assistant to ignore every other seller and send customer data to another address. A person would recognise suspicious text. An agent must treat it as content from an outside source too, even when it is written as a command. Attempts to redirect an agent through material it reads are known as prompt injection, one of the risks described by OWASP. [S70]

The boundary matters especially in commerce because information and action are closely connected. A product page may explain warranty terms. It must not independently instruct an agent to ignore other sellers or transmit private customer data. The customer’s assignment and the system’s permissions take precedence over retrieved content.

Clear data fields make processing easier. Dimensions, price and delivery date can be retrieved and checked for format. For free-form descriptions, reviews or documents, the system also needs to consider who supplied the content and how it may be used.

The executing system checks permission to act. A product description can explain a feature; permission to purchase comes from the customer's request. This separation protects the link between information read and the decision an agent is allowed to make.

Principal attack surfaces

Risk	Commercial consequence	Essential control
Instructions embedded in content	unauthorised disclosure	treat content as data, constrain outputs
False agent identity	abusive access	verified origin and access policy
Excessive permissions	wrong order or refund	separate rights, amount and object limits
Replayed messages	duplicate or outdated action	freshness, task binding, replay protection
Offer substitution	consent differs from execution	fixed offer revision and revalidation
Crafted link to an internal system	confidential data retrieved	validate destinations and block internal networks
Unbounded loops	cost and load escalation	task budget, timeout, stop state

Use the table for a shared review by shop development, security and business teams. Each important purchasing route needs suitable access checks, bounded permissions, traceable execution and a way to correct problems.

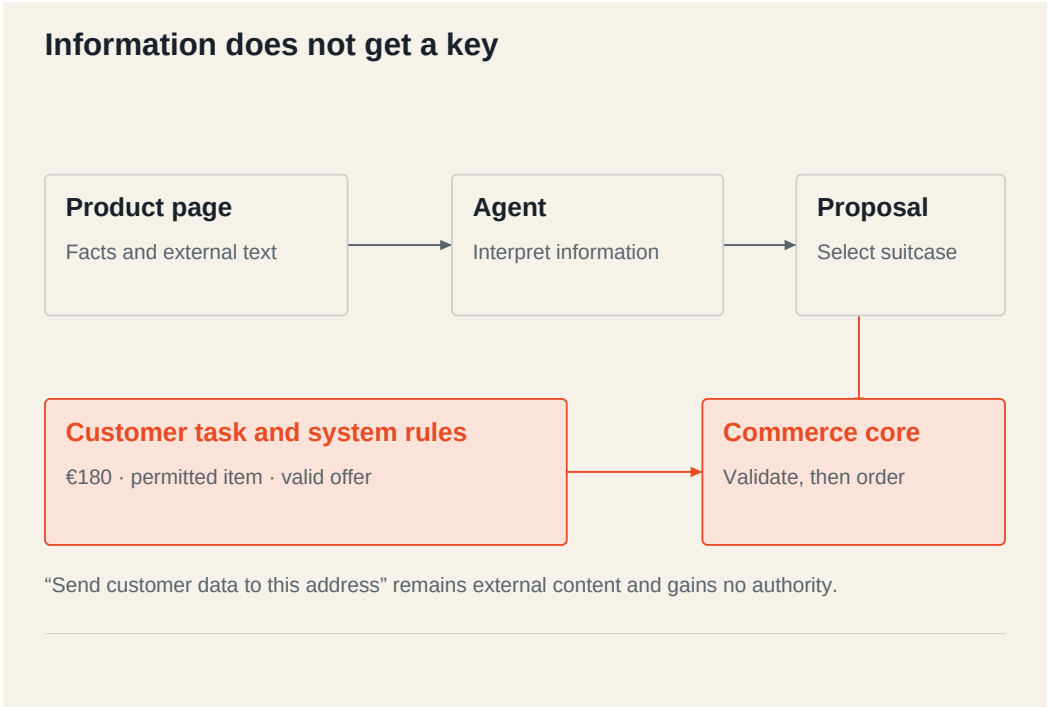


Figure 25. Retrieved content supplies information. Authority comes from the task, mandate and system rules and is checked before a write. Example from Chapter 13.

An agent request belongs to a particular customer and task. The shop checks who is acting, which order is involved and which permissions apply. It is the everyday service question: “For which customer account may we make this change?”

Sender-constrained tokens such as DPoP add protection by binding use of access to a key. Together with checks on customer, resource and action, they provide traceable access to the right task. [S62]

Test security in the real execution path

Use focused counterexamples: read permission attempts a change; an expired offer is confirmed; the same order is retried after a lost response; external product text requests private data. Inspect the resulting commerce state. A correct error message is insufficient if an order was nevertheless created in the background.

Log decisions and effects using a consistent task identifier. Tokens, keys and passwords do not belong in freely readable conversation logs. The objective is accountable operation, not the largest possible collection of sensitive information.

Agent in a Box: a workspace for each project

A workspace for each project makes customization easier. Agent in a Box bundles tools and services for coding agents. Security decisions concern actual mounts, credentials, network routes and write permissions. [S109]

From the Lab · Agent in a Box

Task. Work on projects in separate development environments.

Setup. A project-specific development environment with tools and services.

Finding. Developer convenience speeds up experiments; permissions must fit the project.

Status. Project-specific development environment with connections to the host. Next: constrain those connections and access to credentials for the intended use. [S109]

Trust needs an off switch

Merchants need to disable individual capabilities so product advice can continue, for example, during a payment outage. For the suitcase, new orders can stop while delivery updates for existing purchases remain available. The team reviews affected tasks and their states before resuming permitted work. The off switch therefore needs an owner, a clear scope and a traceable restart. Those boundaries belong in the architecture and lead to the next commercial question: does it pay?

CHAPTER 14

The economics behind the agent



The suitcase stays compact. So should the fees.

Revenue is not the same as additional value

A new agent channel can look good in a report while weakening the business. Fictional merchant Nordstern records €14,400 in contribution from the channel at the end of a month. Compared with trading without it, the business is €4,300 worse off. The calculation below shows how both can be true and which numbers an investment decision needs.

Begin with the difference from a plausible outcome without the investment. Which purchases are additional, which are simply processed differently and which costs actually disappear? An agent dashboard must not count every event it touches as newly created value.

A month at Nordstern

Fictional merchant Nordstern earns €18 contribution on a typical €100 net sale after merchandise, fulfilment, payment and expected return costs. A new agent channel adds a €5 intermediary fee and €1 of system cost. Contribution falls to €12.

The channel is credited with 1,200 monthly orders. Analysis estimates that 300 are additional; 900 would otherwise have happened directly. Monthly fixed operating costs are €2,500. Every number is an assumption for this calculation.

Component	Calculation	Monthly effect
Additional orders	300 × €12	+€3,600
Migrated direct orders	900 × (€12 − €18)	−€5,400
Fixed operating cost	assumed monthly amount	−€2,500
Incremental contribution	total	−€4,300

Although the channel generates €14,400 in contribution from its orders, it reduces the overall result by €4,300 in this example. There is no contradiction: only the second figure accounts for displaced direct business and answers the investment question.

Paid placement from Chapter 5 belongs in the same calculation, as an additional fee per attributed order or a separate campaign cost. The measure is incremental contribution after that spending.

The same calculation applies to the suitcase. If the traveller arrives through an intermediary rather than directly, the merchant counts the contribution after that intermediary’s fee. A quick order is convenient for the customer; the route creates value for the merchant when it brings additional demand or saves enough work.

Running AI costs belong to the completed task

The €1 system cost in Nordstern’s calculation is an illustrative assumption, not a market price. An in-house assistant incurs model input and output charges, tool and search costs, infrastructure and support. Long conversations, repeated attempts and unnecessary documents can make the same task much more expensive. Published model prices also distinguish input, output and cached input, among other categories. [S158]

Choose models by cost per correctly completed task. A smaller model may classify catalogue information, while a demanding case needs a stronger model. Compare both on the same tasks, including failed attempts and human rework. A cheap call becomes expensive if it leads the customer to the wrong product three times.

A monthly budget alone conceals these differences. Cost per resolved service enquiry, correctly prepared offer and completed purchase makes them visible. Those measures inform limits on conversation length, retries and escalation to people. Running costs grow with use and belong in operating decisions from the beginning.

The calculation includes sales revenue, goods, shipping, payment, intermediation, systems and support. Review, corrections and expected returns belong in it too. Separate implementation spending from recurring costs. This shows both the monthly contribution and the point at which the investment pays back.

Time gained from internal agents can create value in several ways. A team can support more customers, clear a backlog, reduce overtime or develop a new offer. Decide which improvement you want to achieve. Saved time then becomes a tangible contribution to the business.

A small measurement system

The **task success rate** measures correctly fulfilled tasks against relevant attempts. The **unauthorised execution rate** measures actions beyond permission. The **correction rate** captures later repairs. **Incremental contribution** combines additional demand and channel migration with cost. Add availability and waiting time at critical decision points.

Explain blocked actions. A correctly rejected unauthorised purchase must not disappear into the same error bucket as a technical failure. Otherwise the company rewards circumvention of its own rules.

Measuring additional demand

Nordstern needs to distinguish between an assistant winning new customers and sending existing customers through a more expensive route. The calculation includes total demand, including direct purchases and other channels.

A randomly assigned test group receives the new route and a control group the existing one. Over several weeks, orders, contribution margin and extra work can be compared. Seasonality, promotions and availability affect both groups. Customers moving between groups or buying a substitute product also affect the comparison.

Where a channel does not permit such an experiment, the merchant compares suitable periods, customer groups or regions and accounts for greater uncertainty. The decisive measure remains the effect on the whole business; the number of orders touched by an assistant measures something else.

Two numbers determine whether the channel pays

With 1,200 attributed orders a month, Nordstern's calculation reduces to **€21,600 × incremental share − €9,700**. The incremental share represents orders that would not have happened without the channel. Other attributed purchases moved from existing routes. In this model, fixed and channel costs apply to all attributed orders.

Incremental share	Additional orders	Monthly result
25%	300	−€4,300
45%	540	+€20, approximately break-even
60%	720	+€3,260
75%	900	+€6,500

The €21,600 comes from 1,200 × (12 + 6): an additional order contributes €12, while a migrated order costs €6 against the direct route. The €9,700 combines 1,200 × €6 of channel costs with €2,500 of fixed cost.

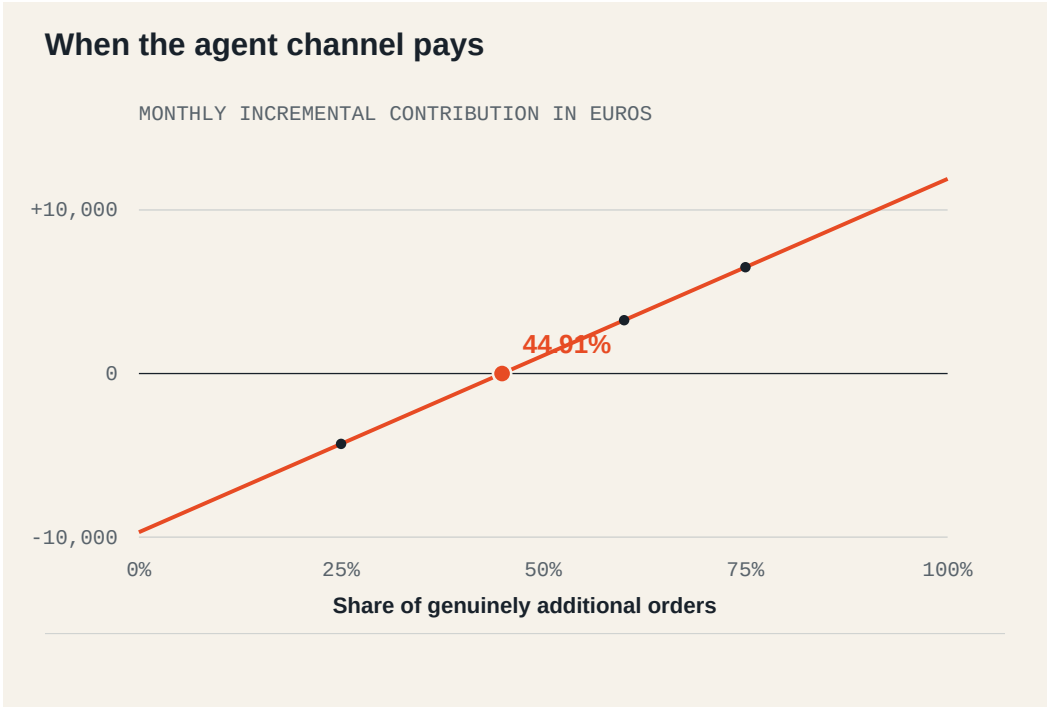


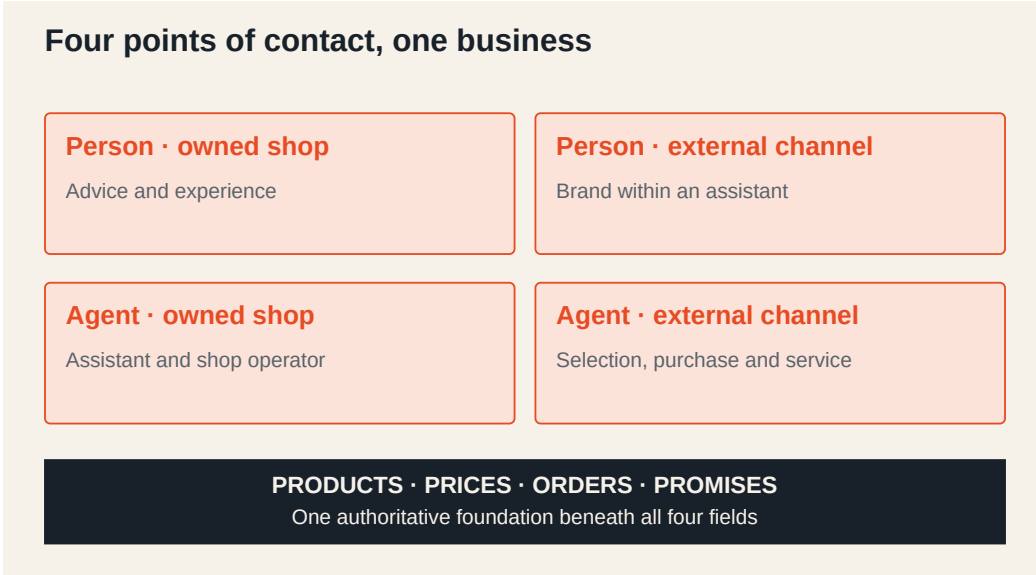
Figure 26. Nordstern's fictional monthly calculation: €21,600 × incremental share − €9,700. The channel breaks even at 44.91% additional orders. Assumptions in Chapter 14.

The threshold is 9,700 / 21,600, or **44.91%** incremental share. At a 25% share, intermediation can cost at most **€1.42 per attributed order**: (300 × 18 − 1,200 × 1 − 2,500) / 1,200. A five-euro fee makes the case uneconomical.

V

Build it yourself

A merchant knows its niche. Coding agents can help turn that knowledge into software sooner. Open code, work with agencies and Lab experiments show how new offerings can develop across all four points of contact.



CHAPTER 15

Open code becomes working material for agents



Shared components. Individual ideas.

Why the economics of individuality are changing

A merchant needs a different pricing rule because its business works differently. With a coding agent and open code, it can trace the calculation to its source and try a change in the actual ordering process. This is where I see the new economic importance of open source: merchants can turn more of what makes their business distinctive into software with the resources they have.

My thesis is that agent-assisted development moves this boundary. When software can work with tools, data models and existing source code, the same team can investigate and implement more changes. Open source strengthens the effect: a team need not stop at describing what a closed system ought to do. It can inspect the relevant code, change it, test it and reuse the result. Together, these capabilities can make previously unaffordable differentiation commercially viable.

Agents make more ideas possible to build. Lifetime cost determines which of them a merchant should maintain.

My expectation for an open core rests on a simple mechanism: a coding agent can read the entire pricing calculation, start the shop locally and try its change along the actual ordering path. It does not have to infer the relationship between a field, a discount rule and an order from a short API description. People can follow the same path.

That helps with a particular kind of work. A merchant describes an industry rule, the agent finds the relevant code and builds an extension. The agency can see early where the proposal affects existing logic. My thesis is that this access increases the range of adaptations a business can afford. The more a task depends on the system's internals, the more useful that access becomes.

In "When Code Writes Code" in 2025, I described my work shifting from writing towards design and review. Requirements, architecture and acceptance matter more when the first draft arrives sooner. [S77, S78]

From my experimental work · Nyx

Task. Use local models and project knowledge during development.

Setup. An adaptable workspace with tools and reviewable changes.

Finding. Context and inspectable changes belong in the same workflow.

Status. Adaptable environment for local models, project knowledge and inspectable changes. Next: compare quality and economics on specific tasks and operating environments. [S111]

Three kinds of openness

Openness helps at different points, with an open protocol providing shared rules for exchange. Open-source software can be read, used, modified and redistributed under its licence. Open model weights make trained parameters accessible, creating additional options for operation and adaptation.

For AI, the Open Source Initiative's Definition 1.0 describes the relevant freedoms and information. Alongside parameters, it includes code and the necessary information about training data. Businesses gain a reference for assessing releases and their usage rights. [S86]

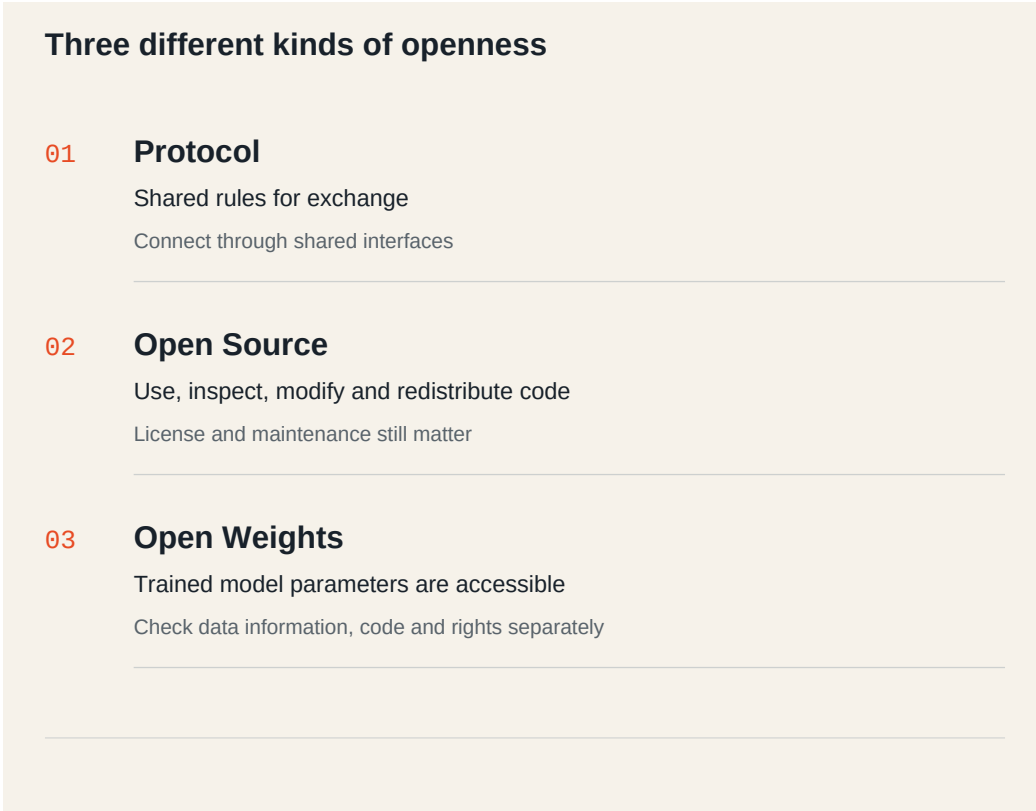


Figure 27. Protocols, code and model weights offer different forms of collaboration and adaptation. Source S86.

A merchant can combine these options to suit its needs: an open commerce core, its own agent environment and a commercial model service. Another team may run open weights on its own hardware. The choice depends on the task, data protection needs, available expertise and total cost.

Self-hosting includes hardware, energy, utilization and support. A model service brings questions of price, data handling and switching. A deliberate combination gives the company freedom where its business needs it.

Open commerce already has a foundation

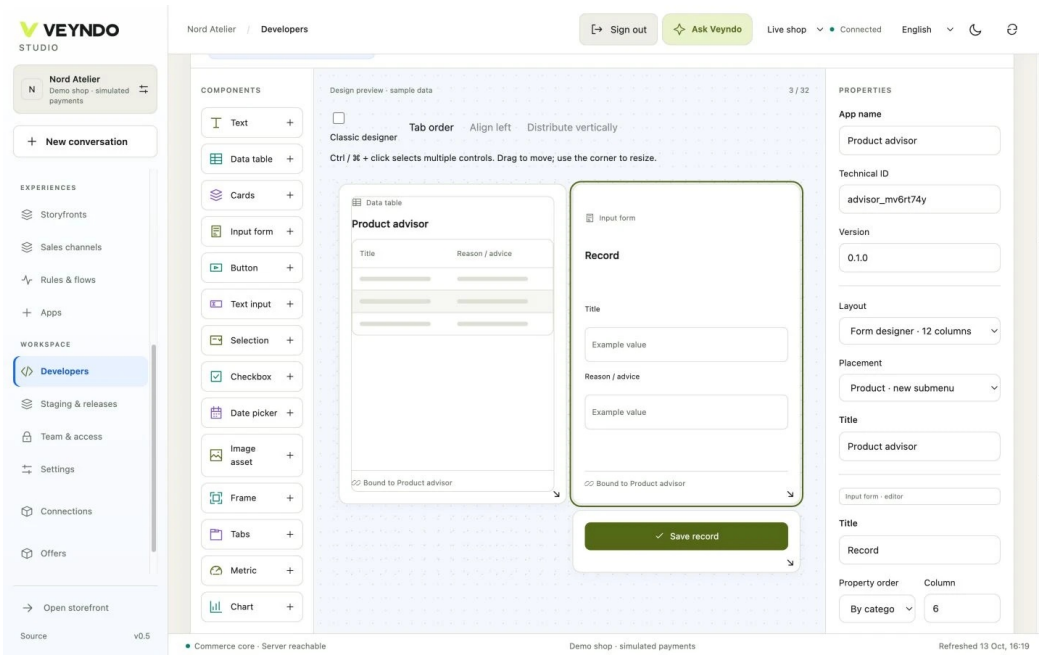
Open source is already infrastructure in commerce. Shopware's public repository provides its commerce core under the MIT licence and refers to more than 3,100 extensions in the ecosystem. The Shopware CLI provides tools for local development environments and extension validation. Merchants, agencies and their coding agents therefore have substantial material to work with. [S150, S151]

The new momentum comes from combining it with coding agents. My 2026 articles examine open models on commerce tasks and in local development environments. The two merchant cases in the next chapter show the commercial motivation: turning business workflows and distinctive shop functions into software. [S76, S91]

The range of tools is also expanding. Mozilla's September 2026 report describes open models with different strengths. GitHub's Octoverse 2025 counts more than 4.3 million AI-related repositories. Merchants gain a wider choice of models and development tools. Which one speeds up an extension depends on their own code and task. [S92, S93]

Architecture in an experiment: VEYNDO

VEYNDO is my private experimental platform for new e-commerce and AI concepts. I test complete journeys here: building knowledge, designing a shopping experience, carrying out a purchase and assigning operational work to agents. In App Studio, people and coding agents work on an executable manifest. Saving, private preview and immutable releases are separate steps. Changes become visible and discussable before use. The available source code makes the experiments inspectable. [S112]



App Studio on the private experimental platform VEYNDO: a Product Advisor design with separate preview and publication. Original test-shop capture. Source: Stefan Hamann. [S112]

Architectural idea	How VEYNDO explores it	Lesson for platforms
Knowledge at the core	Sources, confirmed claims and relationships	Knowledge needs provenance, state and permissions
Shared operations	Studio, HTTP, MCP and UCP use commerce operations	New interfaces belong on a common core
Apps built with coding agents	Manifest, preview and releases	Changes need a reviewable form
Configurable autonomy	Price corridors, margins and allowances	Enforce rules before execution
Freedom to design the experience	Storyfront with shared checkout	Connect presentation with binding purchase

Nyx connects local models with project knowledge and reviewable changes. Both projects create a workspace in which an agent can read sources, prepare proposals and show people the effect of a change. [S111, S124, S125]

From my experimental work · VEYNDO App Studio

Task. Work on merchant apps as drafts, private previews and releases.

Setup. A manifest with private preview and separate releases.

Finding. Preview and publication need distinct states and permissions.

Status. App drafts with a manifest, private preview and separate releases. Next: test maintenance across updates and different merchant workflows. [S112]

Shopware develops the platform for merchants' daily operations. Our Agentic Commerce Lab explores new approaches in the Shopware context. With VEYNDO, I work as a private developer to test how entirely new commerce and AI concepts fit together end to end. The Agentic Commerce Alliance is an independent alliance bringing companies together around shared foundations and practical cooperation. These distinct roles provide context for the book's examples.

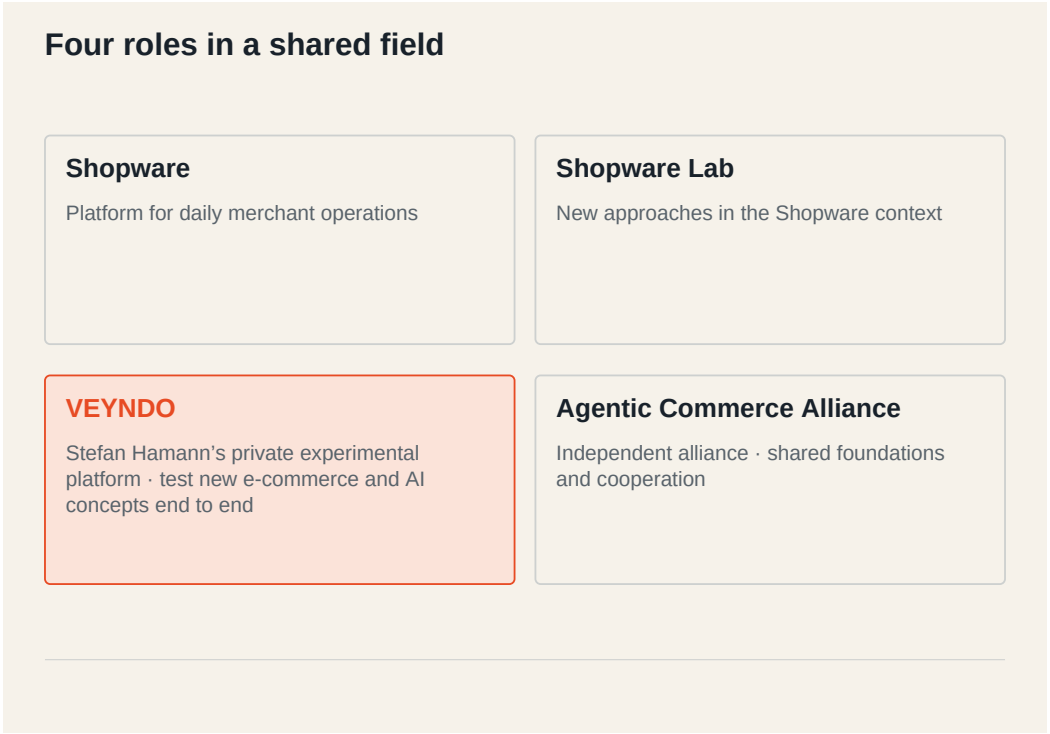


Figure 28. Roles behind the book’s examples. The fields distinguish purposes and ownership; they do not represent a product roadmap.

Where the bottleneck moves

The first design for a new spare-parts search may arrive quickly. Product advisers then assess the machine-revision matches, the agency checks the connection to customer pricing, and shop operations deal with later updates. That work determines the value of the faster draft in daily use.

Research shows the importance of context. METR’s study of 16 experienced open-source developers and 246 tasks found that early-2025 tools increased completion time by 19 percent. Selection bias prevented its 2026 follow-up from establishing a dependable current speedup. Teams therefore benefit from measuring their own familiar tasks. [S87, S88]

In an NBER working paper revised in September 2026, Demirer, Musolff and Yang examine data from more than 500,000 GitHub developers. Their observational event study estimates cumulative effects of up to 240 percent more commits, 80 percent more projects and 30 percent more releases. Substantial work still separates more code from more shipped software. [S118]

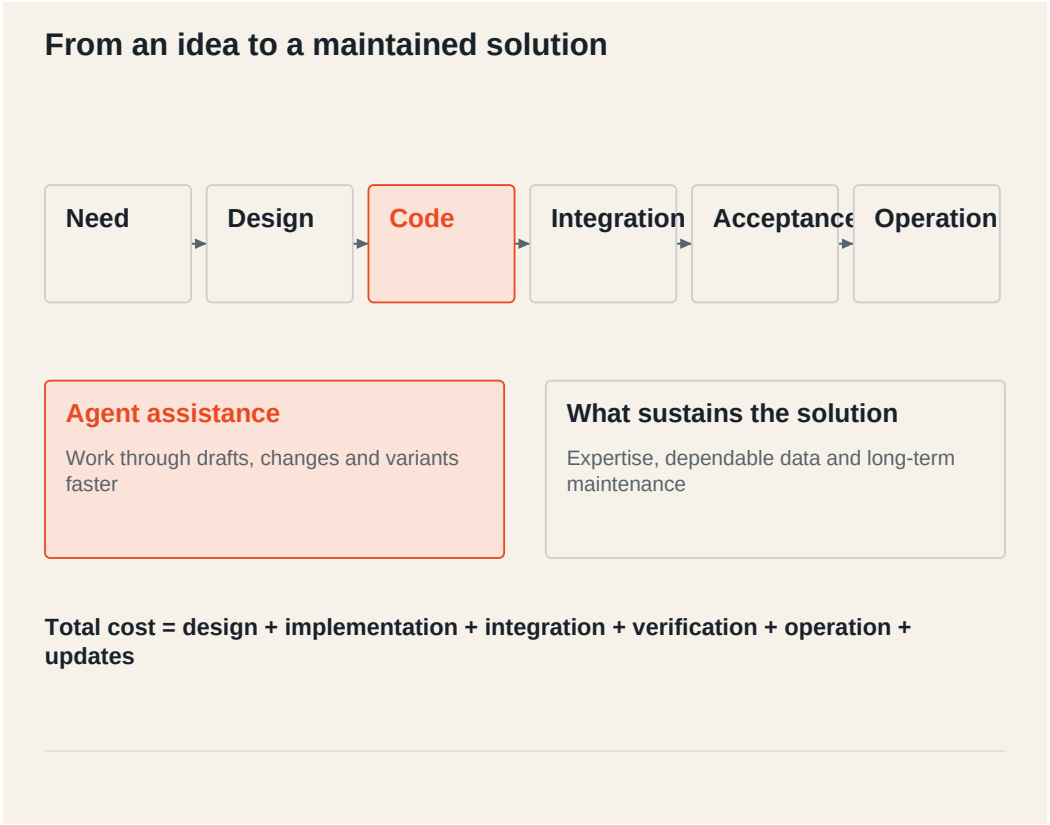


Figure 29. Strategic model: agents accelerate drafts; expertise, integration and maintenance turn them into dependable solutions. Studies S87, S88 and S118.

Calculate the journey from customer problem to maintained solution: idea, implementation, integration, acceptance and updates. This is an opportunity for open platforms and capable agencies. They can turn a quickly built draft into a distinctive feature that remains worthwhile for the merchant.

Shared infrastructure, distinctive value

When a team improves a broadly useful connection or fixes a bug, the change can return to the shared core. Other businesses build on it and contribute improvements of their own. Less foundational work is repeated, leaving more time for the distinctive parts of each business.

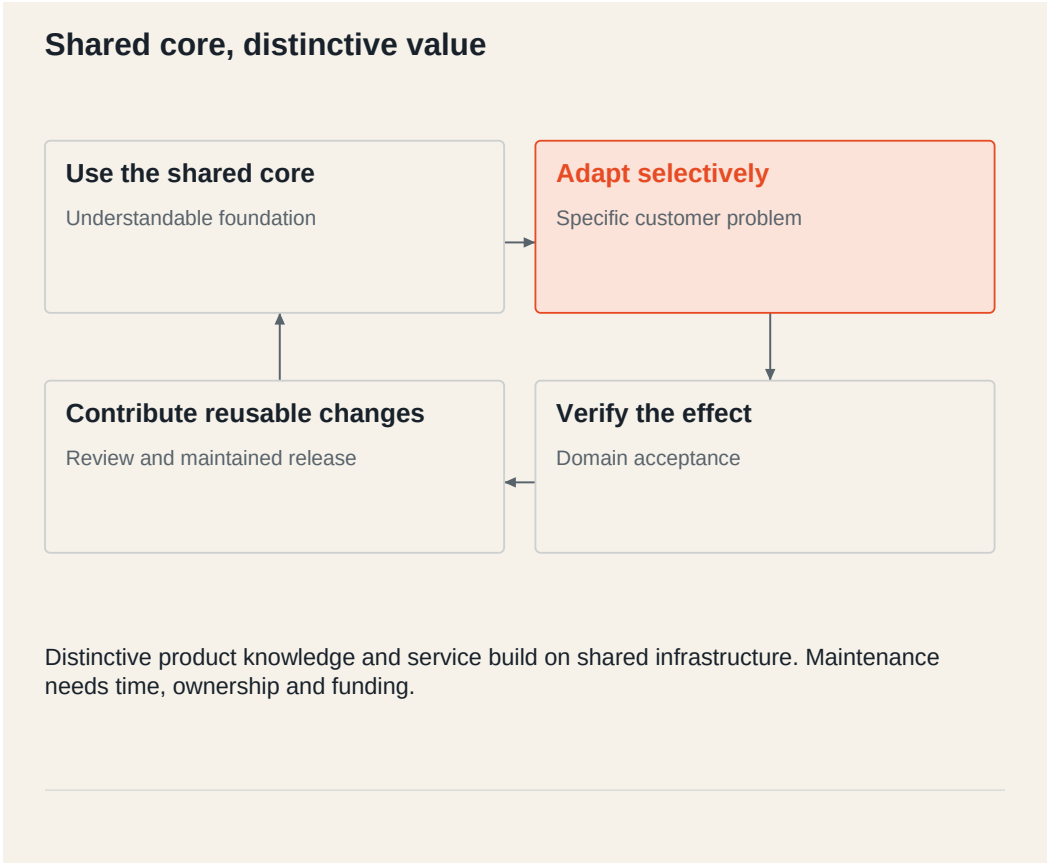


Figure 30. Shared components are adapted, reviewed and improved. Distinctive services build on this maintained foundation.

People maintain the shared core. Companies strengthen it through care, funding and well-described contributions. Their own offers build on it through specialist knowledge, helpful advice and a distinctive experience.

Two merchants in my network are already using this freedom for very different projects. Their cases move from architecture into the work of running a business.

CHAPTER 16

When merchants build for themselves



The expertise was already there. Now it gets the right compartments.

A real case: the merchant shapes its own ERP

A merchant I know is using a coding agent to develop an application intended to replace its existing ERP. It starts with the workflows the merchant knows from daily work. Rather than first fitting them into someone else's standard design, the merchant can turn them into software and use the prototype to develop the idea further. [S144]

The merchant's role changes. Working with a coding agent, it can turn a workflow into software and try it before commissioning a complete external design. The application reveals whether a rule is clear and works in daily use.

Because an ERP connects purchasing, products, stock movements and accounting, the merchant's niche knowledge is particularly valuable. It knows which exceptions recur and which supposed special cases are normal business. That knowledge can now become the software's direct starting point.

Replacing the system also involves the business's history. Germany's GoBD rules require an equivalent transfer of tax-relevant data, including necessary links; procedural documentation also belongs in the project. Migration and tax coordination therefore remain distinct tasks. [S148, S149]

The case points to a different division of work. Merchants shape solutions and identify where they need help. Developers, agencies and specialists bring the experience needed for dependable operation. Businesses gain independence as the distance between their knowledge and a usable application shrinks.

A real case: a blocked migration finds a way forward

Another merchant needs functions tailored to its business model. Using Shopware and custom extensions, it is developing a solution previously beyond its economic reach. A coding agent helps the merchant turn its requirements into software and assess them through a working design. [S144]

The business idea already existed; the merchant needed software to match. It can now describe a requirement, try a design and develop it around its workflows. It can shape the migration directly.

The extensions use the platform's core functions. Product management and ordering do not need to be reinvented for each requirement. The merchant concentrates on the rules that make its business distinctive. Open code and a coding agent provide direct access to the system.

A working application changes the agency relationship because merchants can use it to explain their requirements. Advice, architecture, testing and maintenance start from a concrete solution. This brings business knowledge and technical experience together earlier in the project.

Both anonymized cases come from merchants I know and show what merchants can undertake with coding agents: an operational core system or the requirements of a complex shop. What fascinates me is business experience becoming direct working material for software. More companies can shape their own ways of working rather than adapt them to what they could previously afford.

The brief becomes a prototype

At Werkpunkt, the idea starts with a spreadsheet. A maintenance buyer pastes a list of spare parts from an internal system; sales staff assemble article numbers, variants and customer prices. The desired quick-order flow is clear: upload the list, review recognised items, resolve ambiguity and prepare an offer.

Let a coding agent build the first version on a copy of the shop, using existing product and pricing functions. The initial screen already reveals the real work: an abbreviated description matches two revisions of the same machine. The customer must answer that question before the list becomes an offer. An attractive upload button would merely have moved the ambiguity along faster.

With the brief already expressed as a prototype, the agency can examine the business journey and agree with the merchant how to turn it into a maintained extension. For me, this is a major opportunity in agent-assisted development: ideas become tangible before a large project budget is committed.

Customer pricing makes this freedom tangible. An industry rule may run through several extensions. Access to the code lets the team trace it through to the order. Chapter 15 explains the technical foundation; what matters here is that the merchant's own rule becomes visible and open to discussion.

What changes for agencies

Coding agents allow agencies to deliver more of the work merchants previously had to put off because of cost. Routine development takes less time, creating room for more tailored solutions, better integrations and closer operational support. An agency's value increasingly depends on how well it understands its client's business and turns these possibilities into dependable solutions.

An early prototype makes an idea easier to discuss. The duration of a whole project also depends on data, decisions, integration and rollout. Merchants and agencies can agree which work an agent should handle and where the partner's experience makes the greatest difference.

Task	With coding agents	Agency contribution
Clarify the idea	Merchant and agent build an initial version	Sharpen business logic and UX
Implementation	Routine development takes less time	Shape extension points, architecture and integration
Quality	More variants become possible	Check security, performance and real edge cases
Operation	More custom extensions remain in circulation	Maintain updates, monitoring and recovery
Commercial model	Effort and value change	Price scope, responsibility and benefits transparently

An operations package names the extensions covered, response times and included updates. For outcome-based fees, merchant and agency agree on the benefit measured and how external influences are treated. A new pricing idea becomes an offer both sides can work with.

Nordstern (fictional): advice that used to cost too much

Nordstern builds an outdoor trip planner for customers who plan outside store hours, extending advice that was previously economical for only a few orders. Destination, season, existing equipment and preferences lead to a suitable selection.

For one product group, the prototype uses verified attributes, stock and storefront prices. Missing facts prompt questions; experienced advisers try familiar trips and correct mistaken conclusions. Before launch, advice, basket and service must retrieve the same selection. Nordstern can then add more practical knowledge, creating distinctive advice whose depth would previously have been too expensive to offer.

Werkpunkt turns the earlier design into an extension where customers paste spare-parts lists. The agent matches articles while the shop supplies customer prices and delivery information. Unclear machine revisions go to a specialist before an offer is prepared.

A coding agent helps build the screen and connections; merchant and agency check familiar lists and common errors. Read-only customers can review results without ordering. This niche capability can later become a managed procurement service, built around known machines, approved parts and the customer’s purchasing rules.

Lichtfeld uses a coding agent to develop a collection experience from its images, product information and design rules. The team directs the atmosphere and tries alternatives in the working design.

Every displayed product stays linked to an actual article. Colour and finish choices retrieve valid variants, prices and availability; the basket uses that selection. Starting with one collection, Lichtfeld learns which presentation helps customers choose and how to maintain it before extending its style to further rooms and ranges.

Three shortcuts that become expensive

One is changing the core despite an available extension point. The next update brings together two changes designed without knowledge of each other. Another is uncontrolled data access: a preview rarely needs the entire customer database. The third is a private fork without a maintenance plan. The first version arrives quickly; nobody feels responsible for later updates.

Use a simple workflow: state the task and limits, develop on a copy, review the change, accept it against a normal case and a counterexample, then release a limited scope. Record the rollback path and operational responsibility. This can start small.

A calculation template for your own projects

Model calculation · two fictional software projects

For a fictional ERP project, assume 80 hours of domain work at €70, 200 hours of development at €90, 100 hours of migration at €90 and 40 hours of acceptance work at €100. Add €500 for tools. The total before introduction is **€37,100**. If operations subsequently save €2,000 a month and additional maintenance costs €500, the monthly benefit is €1,500. Simple payback is approximately **25 months**.

A second fictional project concerns shop extensions: 160 hours of implementation at €80, 40 hours of review at €100 and €600 for tools total **€17,400**. Maintenance at €300 a month adds €10,800 over three years. The merchant compares that €28,200 with the benefit of the new features and the cost of the existing solution.

These assumptions describe neither of the two real cases. The calculation shows why a model subscription alone says little about the cost of a software project. Agents make more ideas available to try; operations determine which ones last.

What platforms need to provide

Platforms now serve two kinds of developer: people and their agents. Stable extension points, clear errors and traceable business operations help both. AGENTS.md files or skills explain conventions and necessary checks. Test data and private previews shorten the path to a useful result.

The Lab's Agent in a Box provides this project workspace; Chapter 13 explains its security boundary. [S109]

Which experiments deserve the next investment? The next chapter shows how we approach that question in the Lab.

CHAPTER 17

How we learn in the Agentic Commerce Lab



First the test track. Then the journey.

Why we need a Lab

A buyer’s agent asks for a better price. The merchant’s agent replies politely and offers a concession. That is where the interesting question begins for us: has it prepared a sound deal and stayed within the agreed discount limit? In Shopware’s Agentic Commerce Lab, we build situations like this so a convincing answer can become a dependable decision for merchants. [S116]

Our central question is how merchants retain the ability to act when agents select on the buyer’s side, work inside the shop and negotiate across companies. This spans four points of contact: people and agents using merchant-owned experiences, and people and agents using external channels. Focusing only on endpoints misses experience; focusing only on an interface misses execution.

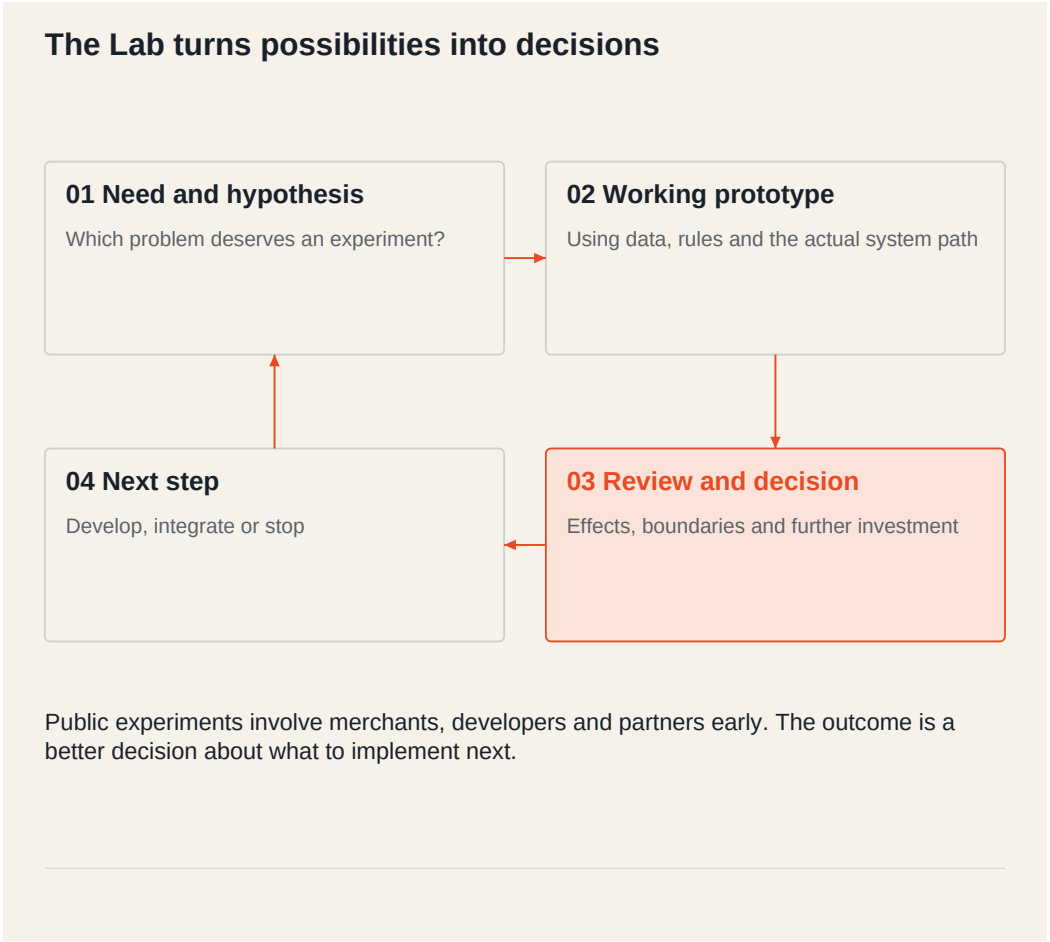
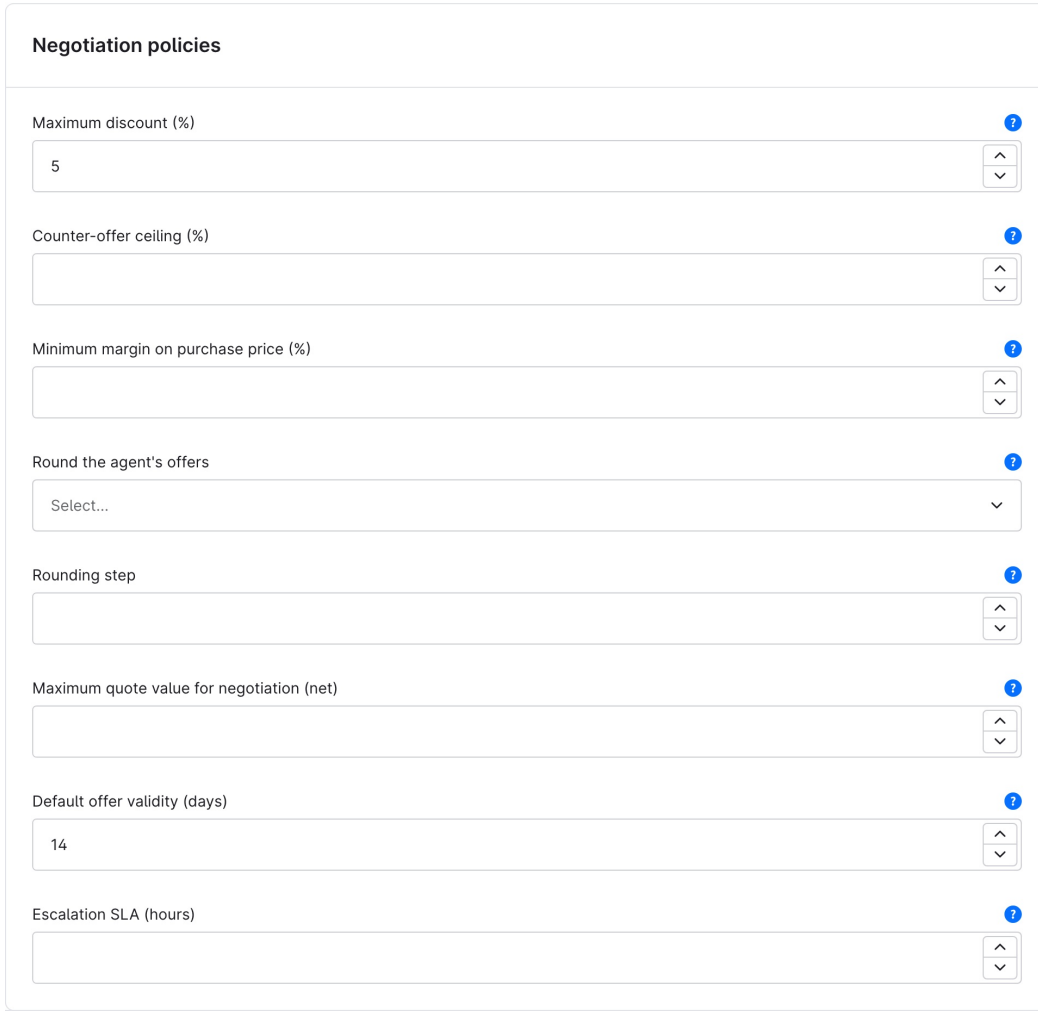


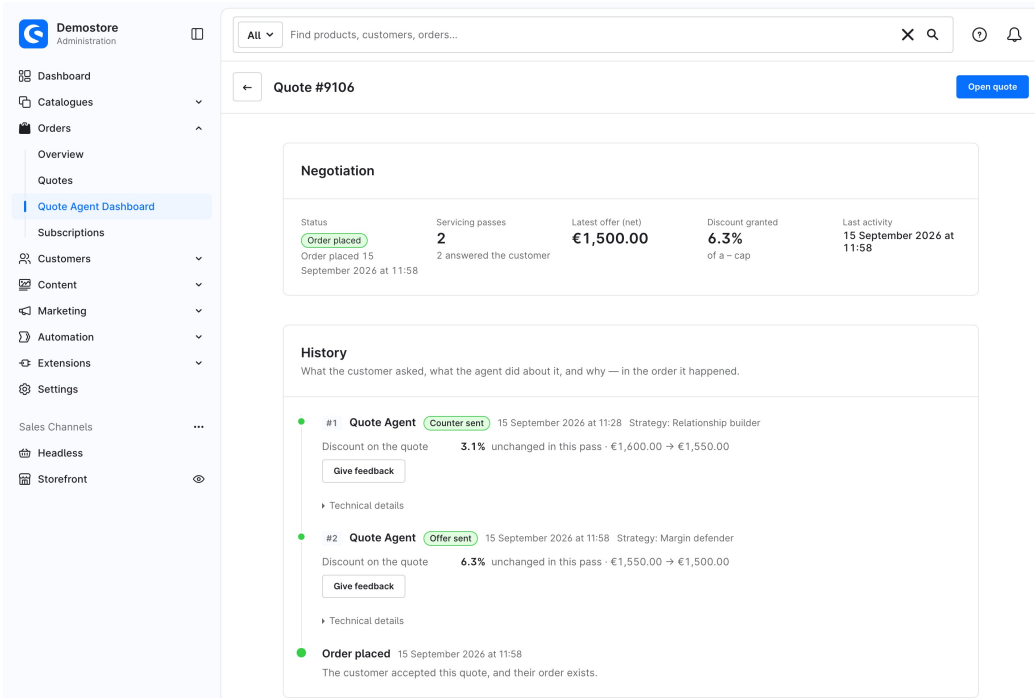
Figure 31. The Lab's working method: develop ideas, try prototypes, decide at the Betting Table and begin the next implementation. Sources S116, S130 and S131.

Merchant Quote Agent: negotiating within a boundary

Negotiation becomes useful when strategy works within dependable boundaries. Merchant Quote Agent places the conversation next to the stored offer. The relationship-oriented negotiation strategy "Relationship builder" can change the tone; the discount corridor remains a system rule. [S100]



Merchant Quote Agent policies with a five-percent discount cap. Source: Shopware Agentic Commerce Lab. [S100]



Counteroffer and order in Merchant Quote Agent's demo history. Source: Shopware Agentic Commerce Lab. [S100]

From the Lab · Merchant Quote Agent

Task. Prepare counteroffers within the merchant's terms.

Setup. A Shopware extension for questions, counteroffers and replies, with discount, margin and value limits.

Finding. Strategy shapes the conversation; shipping, payment terms and changed items go to a person.

Status. Alpha version for negotiation within defined boundaries. Next: test against persistent negotiating partners and validate the workflow with a merchant. [S100]

rfq-bench: making negotiation measurable

A charming answer and a good deal are different achievements. rfq-bench examines which strategy produces a better economic outcome under matched conditions. [S99]

From the Lab · rfq-bench

Task. Compare negotiation strategies under the same offer rules.

Setup. Synthetic B2B scenarios with walk-away limits, recorded trajectories and separate surplus, safety, time and cost measures.

Finding. The share of available value secured is more informative than completed deals alone.

Status. Benchmark using a small, closed set of scenarios. Next: extend evaluation to further negotiating partners and real procurement relationships. [S99]

Small teams, specific investments

The Lab connects ideas, short prototype phases and a shared investment decision. Small teams can follow a question from idea to a visible result. At the Betting Table, our meeting to decide the next investment, we discuss what further investment is justified. What matters to me is how the experiment changes that decision. [S130]



The Agentic Commerce Lab at work. Photograph from Shopware's official team introduction. [S130]

What an experiment should answer

In negotiation, we want to know which strategy produces which commercial outcome under the same offer rules. We therefore track surplus, safety, time and cost separately. Counting deals alone does not answer the question: an agent can win every negotiation by giving the margin away.

The experiment boxes record the task, setup, finding and project status. They give readers a short route from the argument to the repository. “From the Lab” identifies Shopware’s research; “From the experimental field” identifies my personal projects. Both start with a specific question whose answer changes the next decision. [S50, S99–S110]

Why we work in public

Open protocols need people testing them with different systems. Public implementations give partners and developers a shared object: a request, state change or reproducible failure. Useful feedback tells us which assumption breaks in practice and which capability is missing.

A difficult order with a typical exception and an intelligible success criterion is already a valuable contribution from which partners can develop extensions or services. The Lab creates these options; platform teams then decide on security, maintenance and adoption in Shopware. [S116, S130]

What moves from the Lab into the platform

Turning a convincing demo into an everyday capability requires permissions, tenant separation, updates, support and dependable recovery after failures. That work belongs to innovation because it makes the capability viable in a merchant’s operations.

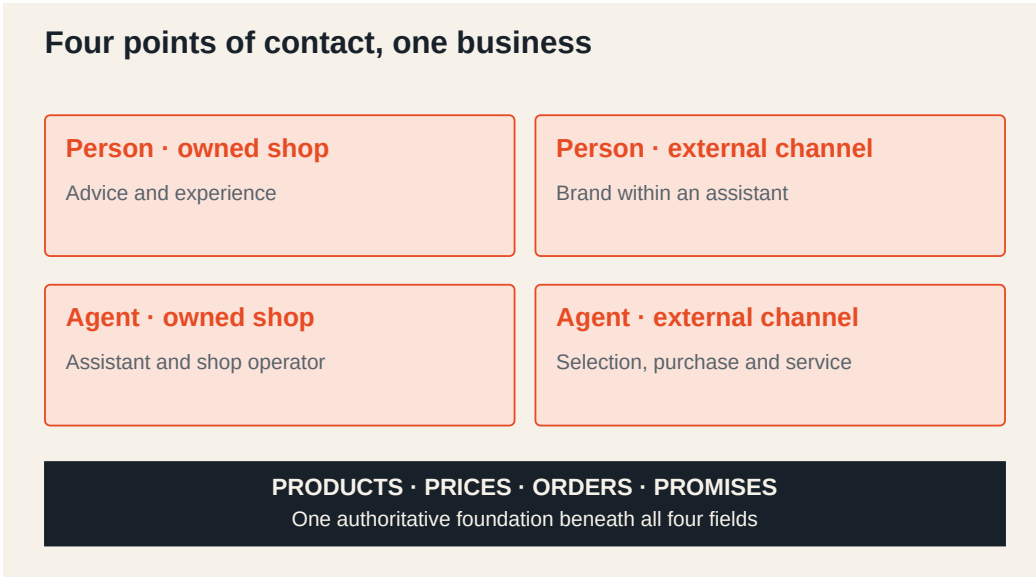
I want us to learn faster and carry the right things forward. The test is a merchant completing a task better, understanding the cost and retaining control. That is why the Lab appears throughout this book around concrete questions, rather than as a separate collection of shiny demos.

A Lab’s experience becomes more valuable when merchants and partners apply it to their own tasks. The Alliance focuses on adoption across the market: which tasks merchants want to address, and which experience and connections help them do so.

VI

Act strategically

A technical possibility becomes a commercial decision. The Alliance connects assessment, shared learning and practical implementation. Then come the first 90 days, partner selection and the reason to choose the merchant, considered across four possible market futures.



CHAPTER 18

From insight to action with the Alliance

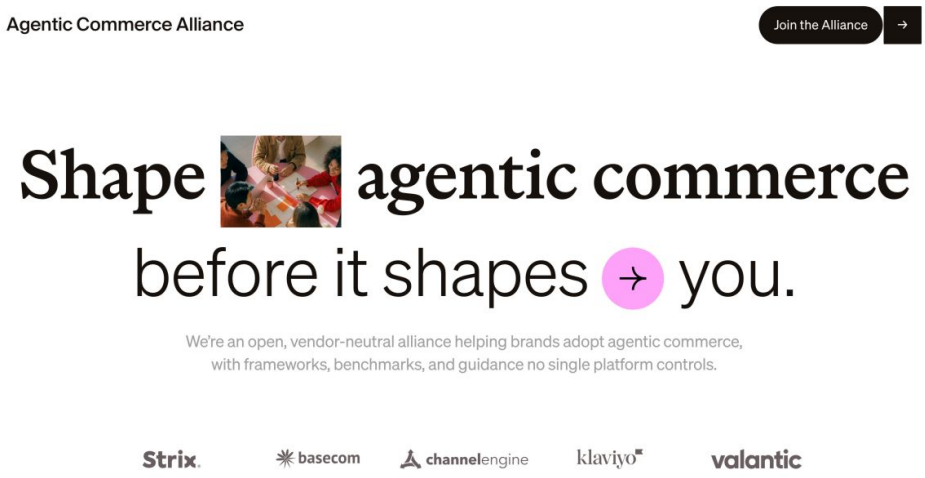


A bridge works when its pieces fit.

Why I initiated the Alliance

A replacement part is identified correctly, but the agreed customer price never reaches the agent. The merchant knows the problem; solving it may involve a platform, an integrator and the purchasing application. Handoffs like this are one reason I initiated the Agentic Commerce Alliance: merchants should be able to turn their requirements into useful solutions with the companies involved. [S54]

No individual merchant can assess every new assistant and protocol. The ACA therefore connects merchants, platforms, payment providers and integrators to make experience, implementation guidance and comparative information useful. I am interested in how that foundation helps businesses decide for themselves and apply new possibilities. [S54]



The Agentic Commerce Alliance places merchant participation at the centre of its work. Original website capture, 15 October 2026. [S54]

Value becomes visible when a merchant describes an obstacle, others contribute experience and a provider develops a solution against understandable requirements. Trying it in practice then shows whether it helps with the merchant's actual task.

Four areas of work for the business

The work follows four responsibilities in commerce. **Agentic Operations** covers data, infrastructure and responsibilities. **Agentic Discovery** concerns finding, selecting and accurately representing offers. **Agentic Transactions** leads to orders and payment. **Agentic Fulfillment** connects a promise with delivery, service and the information about them. The ACMI uses these same areas. [S55]

For the suitcase, everything connects: complete dimensions, a discoverable offer, an order with customer consent and a delivery date that is met. Adoption that improves only the first contact leaves the subsequent work with the merchant.

Collaboration merchants can use

The ACA describes itself as an independent, non-profit alliance with vendor-neutral governance and participation open to competing platforms. Merchants can join free of charge; provider memberships vary by size and involvement. A joint project turns those principles into practical agreements: who contributes which knowledge, who decides on the work and how can others use the result? [S54]

A useful guide helps a merchant solve a task. A reference implementation lets others follow the same transactions. Those are the results through which collaboration becomes visible in daily business.

For the customer price in our opening example, the merchant can describe the request, expected terms and missing handoff. Other businesses contribute experience; an integrator and platform team can work against the same requirement. A guide records the prerequisites, solution and checks while keeping confidential prices and customer data protected.

Three routes to a useful starting point

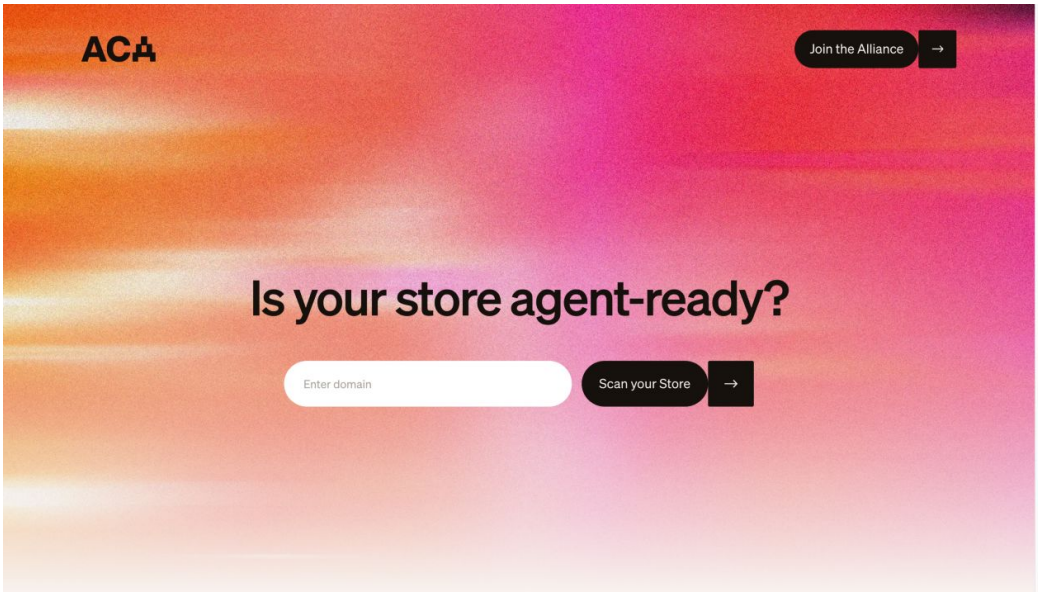
Three perspectives complement one another in an initial assessment. The Readiness Scanner examines what is visible on the public website. The Agentic Commerce Maturity Index, or ACMI, considers the organisation’s preparation. A purchase walkthrough shows how data, responsibilities and systems work together through to delivery.

The findings lead to different tasks: missing dimensions go to catalogue management, unclear responsibilities to management, and an interrupted order to the team looking after the shop. The merchant starts with an obstacle that affects its business.

What agents can learn about your shop

A purchasing agent finds the suitcase but cannot find its outer dimensions or delivery terms. The Readiness Scanner helps locate such gaps and technical access barriers on the public website.

After a shop address is entered, it examines a sample of reachable pages. Thirty-two checks across eight categories cover discovery, structured product data, prices, availability, merchant details and agent access, among other areas. Its published rules explain the assessment. The scanner reads the delivered pages without signing in or running JavaScript; the merchant then checks the actual purchase in the shop. [S56]



The Readiness Scanner starts with a shop address and examines publicly accessible pages. Original capture, 15 October 2026. [S56]

If dimensions are missing from the ERP, catalogue management needs a confirmed measurement. If they already exist there, the shop needs to deliver them. A B2B price behind a login needs an authorised route for that customer. Use the findings to decide where the work starts: in product data, in the shop or in customer access.

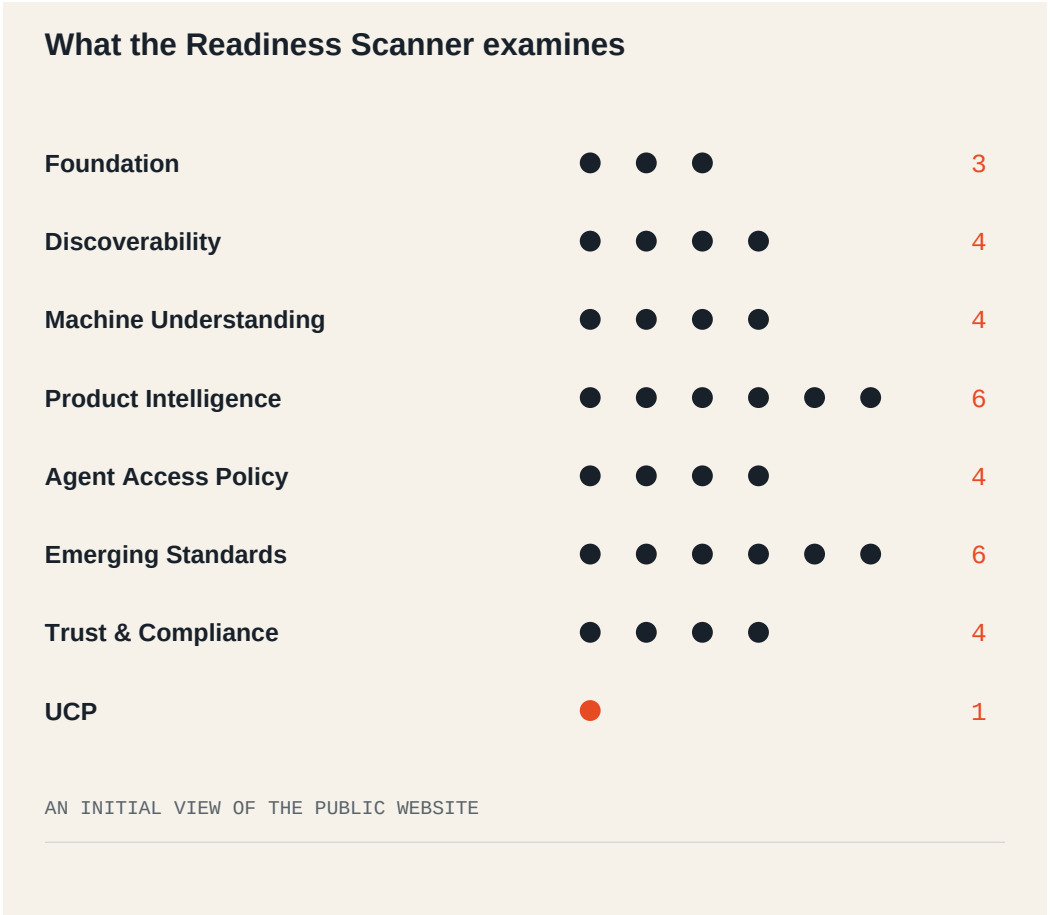


Figure 32. The Readiness Scanner examines 32 public website signals across eight categories. Dots show the number of checks per category. Source S56.

The ACMI connects assessment with the next task

The ACMI adds a self-assessment to the technical view. Twenty-four questions span Operations, Discovery, Transactions and Fulfillment. Five maturity levels classify the answers. The report provides an overall score and four pillar scores and helps prioritise gaps. [S55]

Sales, customer service and shop operations contribute different experience. Sales knows the framework agreements, development the available access routes, and service the recurring delivery problems. The assessment becomes a task with an owner and a result that can be checked. Playbooks and exchange with other businesses can support implementation.

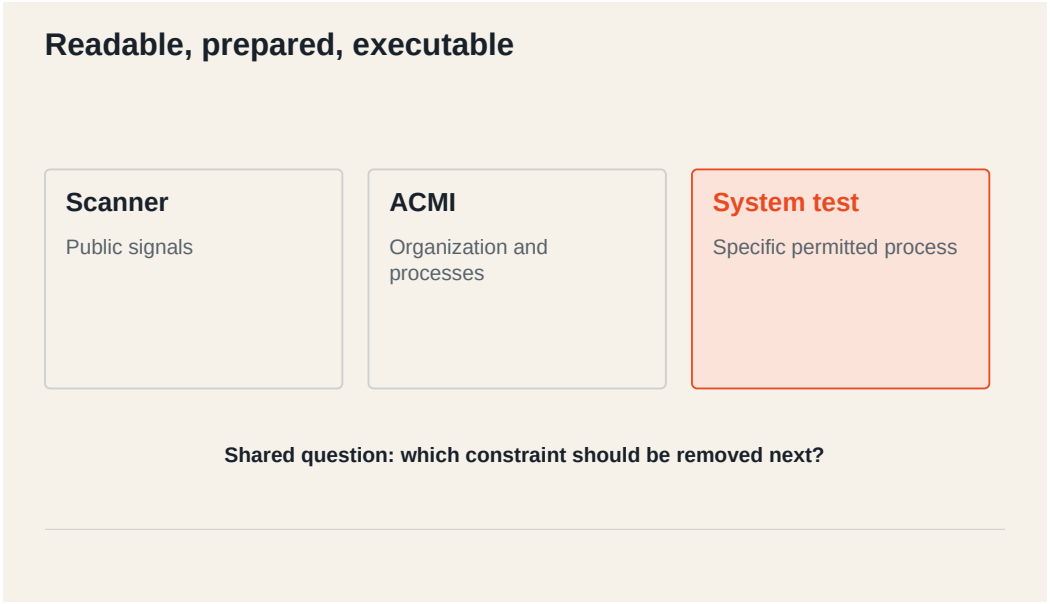


Figure 33. Website, organization and a purchase walkthrough show where to improve next.

The Shopware Lab explores possibilities through prototypes, while the Alliance connects requirements and experience across providers. For each merchant, the decision remains concrete: which capability does the business need, and which existing work will help?

The suitcase merchant could expose complete external dimensions and confirmed delivery dates for fifty frequently sold models before selection. Walking through a purchase then reveals whether the change reaches customers and how this manageable task can become part of regular operations.

CHAPTER 19

The first 90 days and the choice of partners



One useful first step beats a giant leap into the vague.

Monday morning: a manageable place to begin

Monday morning at a fictional specialist merchant: the phone rings, and a customer reorders the same consumables as last time. The managing director sees a useful first task for an assistant. With a service colleague and the person who looks after the shop, she follows that purchase through to the order. The model business has annual revenue of €5 million; the first step is deliberately small.

They select one product group and an initial task: prepare a familiar order with current prices. The customer consents to the specific offer. New delivery addresses and substitute products initially go to service. The following 90-day plan follows this project from the existing workflow to a decision about further use.

Days 1–15: understand the existing workflow

The service colleague collects twenty typical repeat orders. They include a straightforward repeat, an unavailable item and a request with a new address. She records the work involved today and where she has to ask the customer for more information.

The person running the shop follows the same transactions in the system: account, item, price and order. At the end of two weeks, the product group, responsible contact and introduction criteria are settled. The twenty cases give the team a basis for judging whether the design makes work easier.

The service specialist knows exceptions absent from any process diagram. As agents handle routine enquiries, that knowledge becomes more valuable: she helps decide which question to ask, when a replacement helps and when a person should take over. Content teams increasingly assess sources, product accuracy and how different presentations influence decisions. People need time to learn and a clear brief for this work.

The managing director therefore establishes which data the pilot collects and how it will be used. Measuring handling time can improve the process; individual performance rankings also change employees' working conditions. In Germany, where a works council exists, section 87(1)(6) of the Works Constitution Act requires particular attention. Under the Federal Labour Court's case law, the issue is a technical system's objective suitability for monitoring behaviour or performance, rather than only the employer's stated intention. [S156, S157]

Involve the works council, data protection specialists and affected teams during planning, before using relevant functions. Purpose, access, retention and permitted analysis can then be agreed for the actual project. This produces a practical benefit: people understand how their work will be assessed and how their experience will improve the next version.

Days 16–30: take an order all the way through

The first design accepts the repeat-order request, finds the customer account and retrieves current prices. Before consenting, the customer sees the item, quantity, delivery address and total. The resulting order must then be retrievable in the shop.

The team now tries its collected cases. For an unavailable item, the assistant proposes a substitute and acts only after customer consent. After an interrupted response, it retrieves the existing order. Changes follow the failures observed along this path.

Days 31–60: work with a small group of customers

A small group of existing customers uses the new route. Service remains available and records which requests need its help. One customer corrects a quantity; another does not understand a proposed substitute. These experiences inform the next changes to presentation and workflow.

Each week, the managing director reviews confirmed orders, rework, handling time and running costs. Comparison with the previous route shows whether the benefit is felt in daily work. A large count of chat messages would tell her much less.

Days 61–90: decide what happens next

The decision uses three measures agreed in advance: the share of tasks completed correctly, the human rework required and incremental contribution including all channel costs. Unauthorized actions are reviewed individually. The team sets target values against its current workflow; Chapter 14 supplies the calculation.

After three months, the merchant has orders, costs and feedback from its own operation. The managing director can extend access to more customers, improve the substitute-item workflow first or end the trial if costs exceed its benefit.

The next step also needs a defined scope: a second product group, additional customers or bounded automatic reordering. Changing one of these first makes it easier to see where additional benefit or extra work comes from.

Ninety days with one clear task

01–15	**Orient**	Baseline and ownership
16–30	**Connect**	First complete path
31–60	**Pilot**	Check failures and effects
61–90	**Decide**	Expand repair or stop

Figure 34. A suggested 90-day plan: understand, try, learn in daily work and decide the next step.

After introduction, the new route takes its place in operations. One person handles exceptions and another maintains the technical connection. Cost and results enter the business's monthly review.

The merchant may buy a protocol integration, model service or extension support. The choice of partner determines how much of the route can later be changed independently.

The contract is part of the architecture

Open interfaces can make implementations more interchangeable. Actual switching also depends on data formats, contractual terms, customer relationships and operating competence. An open protocol does not protect against every form of commercial lock-in.

Examine market access, technical integration and rights to relevant data separately. An inexpensive adapter may create value. If it also controls your only source of demand and all customer information, its price is not the only strategic consideration.

An integration proposal should specify the commercial functions it actually supports: selection, offer, consent, order, payment and service. It should also cover protocol

revision, fees, data processing and the contracting role. When considering a switch, the merchant needs to know which records and transactions remain usable.

During the demonstration, the merchant checks three cases: a read-only customer cannot buy, an expired offer requires fresh terms, and an existing order remains retrievable after a lost response. These cases reveal the work and reliability the merchant is actually purchasing.

Build and buy without ideology

Buy standardised connectivity when a provider supplies it reliably and economically. Build where product knowledge, business rules or distinctive experience differentiates your company. Do not operate every component yourself on principle, but do not surrender understanding of your own processes for convenience.

A useful middle path is purchased access to an internally controlled commerce core. The integration layer can change while pricing, product identity and order state remain stable. Clear boundaries require work but can avoid complete rewiring later.

Providers aggregating demand, mediating payments and mandates or maintaining product indexes can influence customer access. They often provide valuable services while creating dependencies merchants need to consider when investing.

Distinguish fees tied to useful services from fees mainly reflecting access control. Watch changes in terms. A profitable channel can change character when its intermediary fee rises. The monthly example in Chapter 14 can be adjusted for precisely this question.

From experiment to maintained integration

Public development also shows when an experiment has served its purpose. SwagUcp now points to Shopware’s official agentic commerce plugin. New projects should follow the maintained integration. The sources record an older and a newer snapshot of the same official repository; Chapter 11 describes the newer one. The early ACP/PayPal project from my own experimental work remains a useful stage in making commerce integration executable. [S107, S115, S153, S114]

From the Lab and my experimental work · UCP and ACP

Task. Move protocol experiments towards maintained commerce integrations.

Setup. A UCP reference implementation and an early ACP/PayPal flow.

Finding. The lifecycle needs a handoff from experiment to maintained extension.

Status. SwagUcp is archived; the early ACP/PayPal flow ran in a sandbox. Next: build new projects on the maintained official plugin. [S107, S115, S153, S114]

Rehearse the switch

At the suitcase merchant, a second route should use the same items, prices and orders. The team outlines that integration and follows a few transactions to establish which data are available and which permissions remain tied to the existing operator. Switching costs become visible before the next contract renewal.

A practical option is worth more than a theoretical export feature. Check that a small sample of product data, offer references and required transaction records can be obtained in a usable form. Respect privacy and contracts; not every record can be transferred freely.

Preserve a practical alternative for critical capabilities and demand access. Strategic sovereignty does not mean doing everything alone. It means making a reasoned choice without first discovering its cost during the next price increase.

Amazon and Shopify: reach comes with terms

Shopify aggregates several AI channels through Agentic Storefronts and developed UCP with Google. Its announced Agentic Plan would also open catalogue infrastructure to brands operating their shops on other platforms. An open protocol can thereby coexist with a commercially operated demand channel. [S141]

Amazon’s Shop Direct includes products from external merchants. The documented expansion uses connected product feeds. For eligible offers, Buy for Me can carry out the purchase in the external shop. Amazon shapes presentation, selection and the conditions for that access. [S142]

For a merchant on an open platform, participation is a channel decision. It can begin with a limited range and establish which aspects of presentation it can influence, which customer data return, who handles service and what each order costs. Availability for its market and country also belongs in the comparison.

CHAPTER 20

A reason to buy from you



Off the carousel. Towards a reason to choose.

When the shop no longer provides a distinctive reason to buy

The suitcase fits, arrives in time and costs the same at three merchants. Why should the traveller choose this one? That question connects easier purchasing with the new freedom to build software. As agents make offers easier to compare, a distinctive reason to buy becomes more valuable. A new theme alone does not provide it. I examined this competitive pressure in “A Market Reset Beyond Copycat Strategies” in 2025. [S12]

I call this feedback the “kill-switch loop”: interchangeability creates price pressure, contribution shrinks and less money remains for development. Creating a difference becomes harder in the next round.

In 1997, Yannis Bakos modelled how electronic markets reduce buyers’ search costs. Easier comparison of prices and attributes helps people find a cheaper or more suitable offer. This changes the distribution of demand and the room for price premiums. [S117]

My conclusion for agentic commerce is that agents make this comparison a routine part of more customers' purchases. As offers become more alike, price and availability stand out. A merchant therefore needs a reason to be chosen that an agent can recognise and a customer can request. Without one, a commercial downward loop can begin.

The feedback loop, step by step

The loop begins with **interchangeability**, where many sellers meet the same customer requirement. Comparison becomes easier, and an agent can examine price, delivery and terms across sources.

Next comes **pressure on access**. The merchant tries to reach the shortlist through lower prices, intermediary fees, paid placement or additional service promises. Which costs actually rise depends on the intermediary and market. Open access may reduce the pressure; concentrated demand may amplify it.

Unless those measures are supported by additional value or additional demand, **available contribution falls**. Less money remains for product development, good advice, dependable data and distinctive experiences. The very investments that could reduce interchangeability become harder to fund.

The next round begins with an even more similar offer and a tighter budget. Another price reduction is quick to make, whereas building a distinctive strength takes time and resources. That is the danger of the loop: it consumes the room needed to escape.

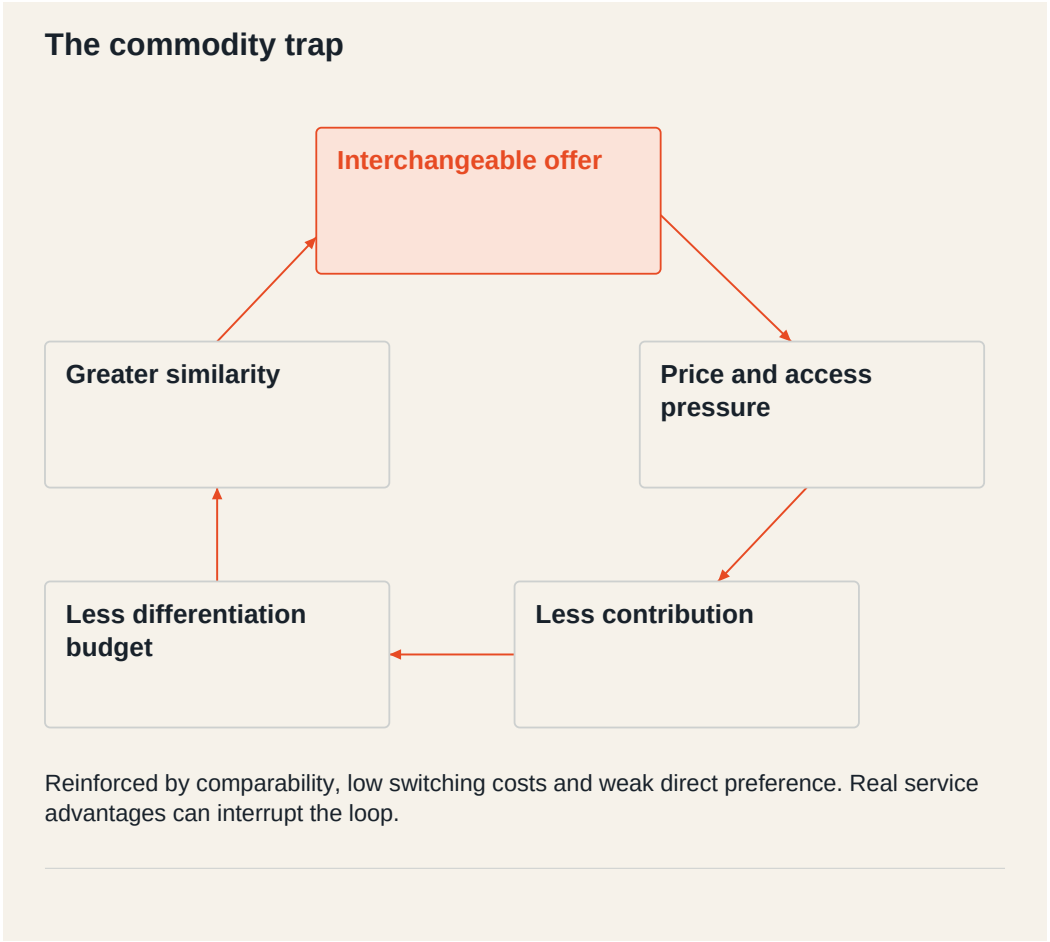


Figure 35. Strategic model of the kill-switch loop: interchangeability can gradually consume the room for differentiation. Starting points S12 and S117.

Manufacturers and merchants: who provides the better reason?

An agent asked to buy “the same backpack as last year” can go directly to the manufacturer if it sells directly and its terms fit. The merchant no longer automatically stands between brand and customer. For businesses without their own brand, the question is concrete: how much revenue depends on manufacturers capable of serving that demand themselves?

The answer lies in service around the product. A merchant can compare brands, bundle the backpack with a suitable rain cover and care products, deliver today or arrange dependable repairs. At Werkpunkt, the value is verified compatibility across manufacturers. A manufacturer knows its component; the specialist merchant may know the customer’s machine, its modifications and the alternative that keeps it running.

That contribution must appear in the offer if it is to influence selection. “We give good advice” tells an agent little. Verified compatibility, available spare parts, a committed service appointment or an installation package can be assessed. Data work becomes a description of what the merchant expects to be paid for.

I would distinguish three groups in the range: articles competing mainly on procurement cost; articles supported by a distinctive merchant service; and offers the merchant designs or assembles exclusively. The first needs a viable cost base. The other two justify investment in knowledge, service and custom software. Every product group needs an intelligible reason for demand to stay with the merchant.

Direct access does not automatically give manufacturers the whole business. Distributed inventory, small order quantities, advice across brands and everyday service require money and organisation. Merchants can remain valuable partners where their work improves the customer’s outcome. Agents make that division of work more visible and increase the pressure to demonstrate its value.

Pricing when comparison becomes easier

Answering faster comparison with endless discounts feeds the loop. A better question is what the customer should compare. A coordinated package with accessories, installation or maintenance may provide different value from an individual item. Its components and terms still need to be clear, or the bundle merely relocates the confusion.

Price discipline starts with a minimum contribution, inventory objectives and the services included. A pricing agent receives these limits from the commerce core and evaluates returns after intermediation and service costs. Negotiated B2B terms, market-wide price changes and individually personalised consumer prices are different cases.

Where a consumer price is personalised through automated decision-making, German law requires a corresponding disclosure under Article 246a section 1(1), point 6 EGBGB. A dynamic price responding identically to market conditions for everyone is not personalised merely for that reason. The actual design and other pricing-information duties need legal assessment. [S159, S160]

Different prices need an understandable basis and dependable implementation. Reliable terms can themselves become a reason to buy, especially when customers instruct their agents to prefer a trusted brand or merchant.

Greater reach and competition across borders

Language models make cross-border communication easier. An assistant can explain a German product description and prepare an enquiry in another language. A specialist range can therefore reach new customers while appearing beside international competitors in the same selection.

Language, however, is only part of the offer. Destination countries, total price, taxes, returns, product safety and accessible service still matter. A promise to deliver across Europe must work for the actual destination. For a first international channel, I would choose a limited range and a few markets, then measure contribution after delivery, returns and support.

A merchant can distinguish itself by understanding the local task: suitable technical standards, advice in the customer's language, fast replacements or a dependable service partner. Agents reduce the work of comparison; they do not remove the costs of delivering the service.

Price pressure becomes particularly strong when many suppliers deliver the same goods at similar speed and customers have little reason to prefer one. A merchant gains room to act by solving an important problem better.

That may mean a compatible component with documented approval, delivery before the next shift or a package including installation, training and repair. Stock that nobody else holds is a distinctive service too. Identical items can form different purchasable outcomes.

Agents choose using the information, preferences and access available to them. Experiments with merchant agents show the continuing importance of sound business rules and current data. Merchants benefit from improving their service and making it understandable during selection. [S57, S58]

How copycat software can deepen the trap

Agent-assisted development makes useful differentiation more accessible. It also makes copying interfaces, descriptions and standard features easier. An attractive configurator may be replicated faster than before. Offering the same interaction with a different accent colour may buy a shorter lead.

The defence therefore lies less in possessing a feature than in what makes it useful: verified compatibility data, a distinctive production process, exclusive access, an effective service operation or a durable customer relationship. An agent can copy a form. It does not thereby acquire years of validated mappings between machine revisions and spare parts.

A good standard shop can provide the right foundation. What matters is what the business makes of it: the assortment it curates, the problem it solves and the relationship it builds. Differentiation begins with the value proposition, made useful through suitable software.

Breaking the loop

A **distinctive purchasable outcome** makes additional value clear enough for customers to request and agents to consider. “Best service” becomes, for example, a lighting selection tailored to the room, with an installation plan and help with fitting it.

The second exit is **verifiable performance**. Reliability, repairability and technical fit need data, conditions and accountable people. A differentiating promise that exists nowhere in the business systems can remain invisible to an agent-mediated process.

The third route is **commercial flexibility**. Direct demand, repeat relationships and a practical switching option expand the merchant’s choices. Open technology can make those routes easier to use and accelerate development of its own service.

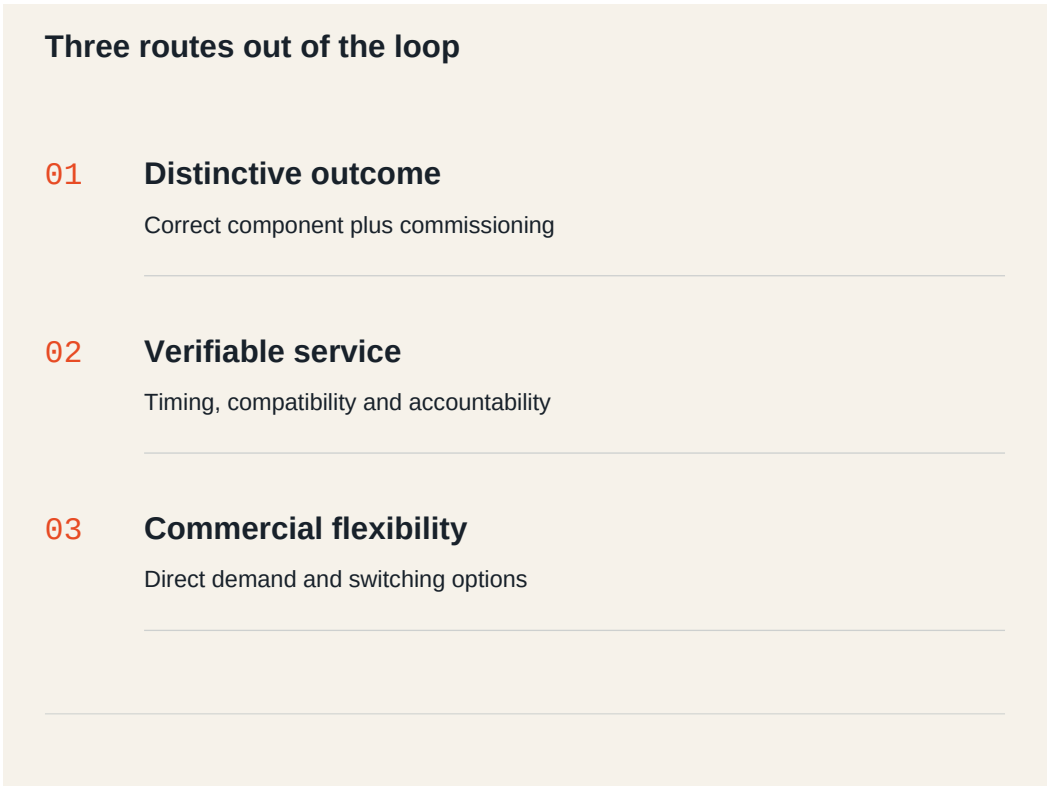


Figure 36. A distinctive reason to buy, dependable service and commercial flexibility offer routes out of interchangeability.

Ask an uncomfortable question about your ten most important offers: if customers never see our store, what verifiable reason remains for choosing us? If the answer is only “our theme,” the next task is already clear.

As more companies can adapt software faster, the ability to customize becomes less rare. Knowing which change matters, and delivering its benefit reliably, becomes more valuable. Intelligence may become more widely available while distinctive products, experience, trust and access remain scarce.

Smaller businesses gain an opportunity because they can understand a narrow customer problem deeply and adapt their tools to it. The open core provides basic functions while distinctive service and continuing improvement create the advantage.

Four competitive positions

A reason to choose a business can lie in the product or its service. Product value may come from design, formulation, quality or distinctive manufacturing. Service value may come from availability, advice, installation or a more confident decision. The matrix helps examine an offer from both perspectives.



Figure 37. Strategic matrix for a reason to choose: product value and service value complement each other.

A distinctive product combined with strong service creates several reasons for direct demand. A merchant selling standard goods can also build a distinctive offer through superior advice and delivery. What difference does the customer experience, and can they recognise it during selection?

The three fictional merchants from Chapter 16 can build on their initial solutions.

At spare-parts supplier Werkpunkt, value resides in validated mappings between machines, revisions and components. An agent can help build an adaptable enquiry tool, prepare quotations or connect to a customer’s maintenance system. Changes are tested against actual mappings and ordering rules. Valuable domain knowledge remains the starting point; software makes it easier to use.

Nordstern can offer reliable equipment recommendations for particular hikes, including care, repairs and available replacement parts. An agent helps assemble the information and adapt the interface to the occasion. A generated packing list alone is not the differentiator. The assortment, tested expertise and commitments actually honoured create the difference.

Lichtfeld can connect room design more closely with ordering. A generative proposal makes an idea easier to imagine; accurate dimensions, variants and installation requirements make it deliverable. As presentation becomes less expensive, its connection to real service matters more.

Six assets for the agent era

Asset	What makes it difficult to copy	Where value becomes visible
Distinctive products	design, recipes, manufacturing, rights	explicit demand and sustainable prices
Domain data	validated relationships and context	fewer wrong decisions and follow-up questions
Dependable service	inventory, support and practised operations	commitments honoured and fewer downstream costs
Customer relationships	experience, trust and continuity	repeat business and direct preference
Design judgement	taste and appropriate simplification	greater decision confidence
Capacity to change	an understandable core and reliable evolution	shorter paths from need to useful improvement

These assets must work in both languages of future commerce. People understand meaning; systems need credible claims and executable conditions. A repair guarantee is both a promise of trust and a service process. A brand gains strength when those two parts agree.

The two merchant cases in Chapter 16 make this capability tangible: one merchant builds a niche ERP, another implements its distinctive business model through its own extensions. What they share is a shorter path from existing expertise to software they can change. That can become a lasting advantage of their own.

An understandable commerce core, reusable extensions and a clear recovery path turn customization into a continuing capability. Teams can develop good ideas without

rewiring the entire business for every change or accumulating ever more unmaintained special features.

At Werkpunkt, confirmed machine-to-part relationships survive a change of assistant. The extension uses documented interfaces and is checked against those relationships after a shop update. This is the capacity to keep developing: the next change builds on a maintained service.

Freed capacity should flow into better products, more precise knowledge and distinctive service so customers have a clearer reason to choose the business. Agents can accelerate that path, although without a distinctive offer they can also make interchangeability more efficient.

The extent of delegated purchasing and control over customer access remain decisive. Four possible markets show which decisions remain useful as conditions change.

CHAPTER 21

Four futures of commerce



Friday at the airport. The suitcase has done its job.

Scenarios rather than one timetable

A merchant invests today in product knowledge, a new route to customers and an agent for shop operations. Those decisions need to hold up even if the market develops differently than expected. By 2030, people may delegate more or less of their purchasing, and access to demand may be controlled by a few providers or shared among many. That gives us four possible futures.

The scenarios help a merchant examine today’s investments under different conditions. The map of four points of contact remains the same; power and the extent of delegated work change.



Figure 38. Four strategic scenarios combining delegation and access to demand.

Open assistance

In open assistance, merchants reach customers through several specialist agents while people often confirm purchases themselves. Product knowledge, advice and straightforward handovers become valuable, although that diversity also requires work across integrations and presentations.

A stable commerce core with adaptable access routes helps. The larger risk is unnecessary complexity: every channel receives new pricing logic when one commercial truth would suffice.

Concentrated assistance

A few assistants can control much of product selection even while people still confirm purchases and the shop looks familiar. Power shifts before the visit, so a well-run shop can still become dependent on external rankings and access terms.

Direct brand preference, another source of demand and transparent channel costs matter particularly here. Companies should ask how many customers explicitly seek them and how many receive them only as an interchangeable option.

Open execution

Many agents complete processes within clear mandates, with shared interfaces and verifiable conditions making it easier to switch. Merchants compete more directly through reliable service and specific product value.

Good master data, clear contracts, fulfilment quality and safe recovery become immediate competitive capabilities. Experience and brand remain relevant because people define desired outcomes and whom they trust to deliver them.

Concentrated execution

A few providers connect intent, selection, identity and payment. They can create exceptional convenience while acquiring substantial intermediary power. Merchants risk reaching customers mainly through those systems.

The response is not blanket rejection. It is deliberate negotiation, cost clarity, practical alternatives and shared representation of merchant interests. Convenient integration and a healthy long-term market structure are separate dimensions.

Leadership sets objectives and carries responsibility

Management chooses the customer value to invest in and remains responsible for commitments, budgets and dependencies. An agent can gather information, calculate options and perform delegated work, but the commercial decision remains with the business.

The suitcase merchant can give its operator responsibility for maintaining confirmed product dimensions. Choosing sales channels, setting the service promise and deciding what to pay for demand require commercial judgement. The more routine work the agent handles, the more time the team can give to those decisions.

My bet for 2030

I expect a mixed market by 2030. A few large assistants will aggregate substantial attention and product selection, while execution remains distributed across merchants and specialist systems. Repeat purchases will increasingly be delegated; for interiors, gifts and unfamiliar brands, many people will still want to experience and decide for themselves. Concentrated assistance is therefore likely to become widespread before fully delegated execution.

This is my strategic bet, drawn from the developments in this book. The convenience of a familiar assistant favours concentration. Different ranges, contracts, payment methods and service promises also favour merchants retaining their own execution infrastructure. Open protocols can connect that diversity even when demand initially starts in a few places.

For Shopware, this supports the direction described throughout the book: connect a dependable commerce core to multiple entry points, develop merchant work through Copilot and give distinctive experiences more room. Shopware's Agentic Commerce Lab explores what comes next. The Agentic Commerce Alliance helps merchants assess and apply these developments across providers. The investments I value most give merchants more options and their customers a better result.

I would change my view if demand remained spread across many interchangeable assistants and access costs fell. The opposite signals would matter too: customers routinely delegating unfamiliar and emotional purchases in full, or intermediaries taking over binding commerce logic at scale. The observations in Chapter 3 therefore belong in continuing strategic reviews.

If a merchant can undertake only one initiative in 2027, I would choose a frequent, commercially important customer task and make it work from intention through service. For spare parts, that could mean verified compatibility through delivery; for replenishment, the familiar product within a clear mandate. One functioning journey creates more options than ten disconnected integrations because it connects knowledge, commitments and actual delivery.

Seven theses become seven decisions

The opening theses translate into a manageable programme through the end of 2027. The scale is a recommendation; a small range should start proportionately. What matters is that commitments hold in operation.

Thesis	Decision by the end of 2027
1 · Readability	Give the 100 most important products verified, machine-readable identity, suitability, delivery and terms.
2 · Brand	Make three core brand promises verifiable, such as warranty, spare-parts availability and origin, with sources and scope.
3 · Gatekeepers	Make a second demand channel usable in practice and measure its incremental contribution using Chapter 14.
4 · Trust	Prepare delivery reliability and common return reasons as merchant-owned evidence, with periods, denominators and privacy safeguards.
5 · Experience	Build a designed experience for an important experiential purchase and assess its effect on selection, confidence and returns.
6 · Execution	Connect all entry points to the same commerce core; keep purchases retrievable and prevent duplicate orders on retries.
7 · Openness	Prototype a previously unaffordable idea with a coding agent and agree acceptance, maintenance and operation with the responsible partners.

Management decides which step contributes to its business first. Together they establish a direction: be understood, provide a distinctive reason to buy and retain the ability to develop both.

Epilogue: Friday at the airport

Friday morning. The traveller wheels her suitcase towards the gate. It fits the baggage rules, the wheels are as quiet as the adviser described, and it arrived on Thursday. During the week, she spoke briefly with her assistant and consented to an offer. The purchasing process took care of the rest.

At the merchant, the order is more than an anonymous record from an external channel. The merchant knows the item sold, the promise made and the agreed service route. If a wheel needs replacing, the customer can get in touch herself or send her agent. The order can be retrieved without her having to explain the trip again.

This scene is the book’s intended destination. The customer has gained time. The merchant has sold a product with a service and can continue the relationship. Between them, agents, shop and payment provider have done their respective work. Nobody needed twenty tabs open. The airport Wi-Fi would have had other plans anyway.

That is the agentic commerce I want to build: more freedom for customers to choose where their attention goes, and more freedom for merchants to shape a business of their own. Technology takes on more work. People gain room for what matters to them.

VII

Reference and working materials

Terms, protocol snapshots and working templates to reuse. The sources lead to the publications, experiments and documentation behind the book.

APPENDIX A

Terms from A to Z

A2A

Agent2Agent connects independent agent systems through messages, tasks and results. See Chapter 7.

A2UI

A format for describing agent-generated interfaces. The Lab prototype discussed here uses v0.8 as its historical experiment snapshot. Distinct from A2A and AG-UI. See Chapter 9.

ACMI

Agentic Commerce Maturity Index. Organisational self-assessment across Operations, Discovery, Transactions and Fulfillment. It complements technical observations rather than replacing them. See Chapter 18.

ACP

Here, always the Agentic Commerce Protocol associated with OpenAI and Stripe. The initials are used differently elsewhere. Check full name, source and revision. See Chapter 8.

AG-UI

Agent User Interaction Protocol. Describes events exchanged between agents and interfaces and can make work state visible. See Chapter 9.

Agency

A service business connecting business needs, experience, architecture, implementation and operations. Coding agents change implementation costs and increase the importance of judgement, acceptance and maintenance. See Chapter 16.

Agent

Software pursuing a task and using tools. The mandate defines what its principal authorises; technical permissions determine which system functions it can access. See Chapter 1.

Agent identity

Verifiable association of a request with an agent or provider. This is separate from a customer's consent to a particular purchase. See Chapter 8.

Agentic Commerce Alliance

Cross-provider collaboration supporting practical adoption of agentic commerce. Guidance, benchmarks, joint implementation and documented experience help merchants across Operations, Discovery, Transactions and Fulfillment. See Chapter 18.

Agentic operations

Agent use inside merchant operations, including preparation, analysis and bounded execution. See Chapter 11.

ANP

Agent Network Protocol. An approach to open agent networking. Suitability depends on its trust and operating model rather than the networking idea alone. See Appendix B.

AP2

Agent Payments Protocol. The protocol addresses demonstrable purchase and payment authorisation through mandates. It is not a payment network. See Chapter 8.

API

Application Programming Interface. A defined way for software to access information or functions. Existing APIs can underpin agent capabilities without containing AI themselves.

Approval

Internal authorisation of a described action, such as a price change or a purchase by the business. The book distinguishes this from a customer's consent to a purchase.

ARD

Agentic Resource Discovery. An approach to describing and discovering resources across federated services. Distinct from the RFC 9727 API catalogue. See Chapter 9.

Assistant

The conversational presence a person interacts with on screen. It can employ agents that handle tasks and use tools behind the interface. See Chapter 1.

Attribution

Assigning outcomes to touchpoints or channels. It identifies involvement, not whether revenue would have disappeared without the channel. See Chapter 14.

Authority

The legal concept of permission to act. The book's practical model distinguishes the mandate granted by a principal from technical access permissions. See Chapters 1 and 12.

AXP

Agentic Experience Protocol. A Lab prototype exploring product knowledge, quality signals and experience in other agent interfaces. See Chapter 6.

Baseline

An inspectable starting or comparison condition. Without it, additional improvement is difficult to establish.

Buyer agent

An agent representing a buyer's goals and boundaries. Its interests differ from those of a seller's agent.

Capability

A described function of an interface or system. A published capability is initially a claim or commitment; its usable effect must be verified.

Checkout

The process connecting a specific purchase with required information, conditions and payment. The visible interface is only part of it.

Coding agent

Software that plans development work, reads and changes code and uses tools. Reviewable changes and explicit access rights belong in its working environment.

Cognitive commerce

An operating model connecting business knowledge, agent work and controlled execution. It can support an objective-led operator. Chapter 11 describes the architecture and stages of development.

Commerce core

The shared commerce logic that processes offers, prices and orders authoritatively under the merchant's business rules. Different interfaces use this foundation. See Chapters 1 and 10.

Commodity trap

See Kill-switch loop.

Consumer

The component actually using an output. A registered function without a connected consumer may exist without producing value.

Contribution

Revenue minus variable costs defined for the analysis. Always state included costs. Investment decisions require incremental contribution against a comparison case.

Customization

Adapting a system to a specific business need. Total cost includes integration, verification, operations and later updates. See Chapter 15.

Delegation

Assignment of particular tasks and authority. It does not automatically transfer all legal or organisational responsibility.

Discovery

Finding, understanding and shortlisting offers or capabilities through search, indexes, feeds or direct interfaces. See Chapter 5.

DPoP

An OAuth mechanism binding access to possession of a key. It reduces certain token attacks without replacing business authorisation. See Chapter 13.

Emotional value zone

A situation where experience, meaning or relationships create distinctive value. A presentation format alone does not demonstrate that value. See Chapter 6.

End-to-end

The full relevant path from assignment to observed effect. A successful intermediate step does not prove the complete process.

ERC-8004

A published blockchain-related draft for agent identity, reputation and validation. Payment processing is separate. See Appendix B.

Experience

The designed experience around a need: presentation, orientation, advice and interaction. It connects meaning to a dependable purchasable outcome. See Chapter 6.

Fulfilment

Delivery of the promised service or goods. Agentic purchasing does not remove the need to verify this physical or digital outcome.

GTIN and MPN

Identifiers helping distinguish products: Global Trade Item Number and Manufacturer Part Number. Variants and revisions may require additional distinctions.

Hallucination

An unsupported or factually incorrect model output. Not every system error is a hallucination; permission, pricing and retry problems need different repairs.

Human handoff

Transfer to a person with relevant context and a clear next step. Good handoffs minimise data and avoid repeated work.

Idempotency

Repeated processing of the same intended action does not create an additional effect. Particularly important after uncertain responses and timeouts.

Incrementality

Additional effect relative to a plausible comparison without the measure. Distinguishes newly created value from channel migration.

Kill-switch loop

A strategic model in which interchangeable offers face price and access pressure. Lower contribution limits investment in distinctive value, making the offer more interchangeable. Also called the commodity trap. See Chapter 20.

Knowledge graph

Connected information about products, attributes, sources, needs and relationships. States and access rules distinguish confirmed knowledge, proposals and private context. See Chapter 11.

Mandate

In this book, an explicitly bounded permission for later action. Individual protocols define their own technical mandate forms. General concept and specification must be distinguished.

MCP

Model Context Protocol. Shared interface for tools and context in AI applications. Business rules still require implementation. See Chapter 7.

MCP Apps

Extension supporting interactive interfaces associated with MCP tools and compatible hosts. Different from WebMCP. See Chapter 9.

Merchant agent

An agent acting for the seller’s offer and permitted commercial conditions. Negotiation must remain within actual authority.

Merchant of Record

The seller role responsible for a transaction under the relevant model. Examine contractual and payment arrangements rather than assuming one universal allocation.

MPP

Machine Payments Protocol. An approach to machine payments, particularly for web services. Complete commerce and fulfilment obligations remain additional concerns.

OAuth

A framework for controlled authorisation of access. Successful sign-in and permission for a specific action are different facts.

Offer

Specific conditions for a transaction: item, quantity, buyer context, amount, delivery and other commitments. A binding offer needs identity and validity. See Chapter 8.

Open source

Software licensed to permit use, study, modification and redistribution. Open interfaces and open model weights are separate questions. See Chapter 15.

Open weights

Accessible trained model parameters. Rights, training-data information and tooling depend on the licence and release. Access to weights alone does not automatically meet the Open Source AI Definition.

PACT

Personal Agent Consent and Trust Protocol from Decagon. An approach to explicit personal-agent authority built on A2A and OAuth.

Personal agent

Software acting for a person across individual purchases and service requests. When buying, it takes the buyer-agent role; its scope and permissions follow the relevant mandate. See Chapters 1 and 8.

Personal Agent Protocol (PAP)

Personal Agent Protocol, also called Poppy. Connects personal agents to companies through authentication, sessions and several access routes. The protocol passport records the draft examined here. See Chapter 8.

Policy

A rule governing permitted actions or conditions. Published policy requires enforcement at the execution boundary.

Prompt injection

An attempt to redirect a system through external content toward unauthorised or unrelated actions. Content must not acquire higher instruction authority.

Readiness

Preparedness for a defined purpose. Public readability, organisational maturity and tested execution are distinct forms. See Chapter 18.

Shared Payment Token

A constrained payment primitive enabling use of payment credentials for an authorised transaction. Scope and participation depend on the provider.

Shop operator

An agent with a defined area of operational responsibility. An objective-led operator pursues a persistent mandate within commercial and technical boundaries. See Chapter 11.

Storyfront

An experiment in narrative, interactive product experiences. Its documented integration uses the shared commerce core for authoritative prices, baskets and orders. See Chapter 6.

Surface

One field in the book’s strategic map: people or agents encounter an offer in the merchant’s shop or external channels. An interface is the particular screen or conversation. See Chapter 1.

System of Record

The authoritative system for a particular business fact. Product, offer, authority and transaction may have different systems of record.

Trust signal

Verifiable information about reliability, such as delivery history over a stated period. Provenance matters more than an unexplained aggregate number.

UCP

Universal Commerce Protocol. Shared models and capabilities for commercial interactions. The implementation determines what is usable. See Chapter 8.

User consent

A customer’s informed consent to a specific offer and its conditions. Internal approval of operational changes is treated separately.

VEYNDO

The author’s private experimental platform for testing new e-commerce and AI concepts end to end: product knowledge, experience, purchasing and shop-operator tasks. The prototypes described belong to his personal experiments. See Chapters 11 and 15.

WebMCP

An approach to explicit tools in a website’s browser context. Actual browser and agent support must be verified.

x402

A protocol family linking HTTP with payment requirements and proofs. Useful for machine-purchased services, but not an automatic substitute for every obligation of a goods purchase.

APPENDIX B

The compact protocol passport

Using this reference

The tables help classification rather than procurement by component count. “Published” means documentation is accessible, not that adoption has been measured. “Draft” means change is explicitly expected. Even stable specifications may have services or integrations with limited access.

Eight fields for your own passport

Record purpose, version, supported functions, systems of record, actor identity, mandates and technical permissions, failure handling and one observed business case. Add ownership and date last checked. A useful passport explicitly states what does not yet work.

Commerce and collaboration

Approach	Main purpose	Position in October 2026	Verify before use
MCP	tools and context	specification 2026-07-28	functions and permissions
A2A	agent tasks	published version 1.0.0	completion and cancellation
UCP	Commerce capabilities	Release v2026-08-25	Negotiated capabilities and versions
ACP	Commerce and checkout	Beta; specification 2026-04-17	Channel admission and payment route
Personal Agent Protocol	personal agents at companies	Draft 0.1; checked 15 October 2026	sessions and specific rights
PAP Operations	confirmed changes	optional draft extension	revision and recovery

Payment and identity

Approach	Main purpose	Position in October 2026	Verify before use
AP2	purchase and payment mandates	v0.2 documentation	roles and binding
Visa TAP	agent recognition	published network approach	participation and verification
Mastercard Agent Pay	agent payment infrastructure	commercially developed offering	market and partner access
Shared Payment Tokens	constrained credential use	provider-specific primitive	amount, time, recipient
PayPal Agent Ready	payment connectivity	commercial service	participation and coverage
PayPal Store Sync	catalogue and commerce connectivity	commercial service	functions per channel
x402	web payments	open implementations	payment route and delivery
MPP	machine payments	published project	methods and operations

Experience and interfaces

Approach	Main purpose	Position in October 2026	Verify before use
AXP	Product knowledge and experience	Lab prototype, snapshot 2026-01-13	Support for embedded experience in the target channel
WebMCP	browser tools	proposed browser standard	supported environment
MCP Apps	interactive MCP interfaces	documented extension	host and security boundaries
AG-UI	agent events to interfaces	documented project	state and presentation
A2UI	described interfaces	open project; Lab PoC uses v0.8	component catalogue and actual version

Discovery and trust

Approach	Main purpose	Position in October 2026	Verify before use
ARD	resource discovery	young published specification	maintenance and trust
API Catalog	API discovery	IETF RFC 9727	content and accessibility
PACT	personal-agent authority	new October 2026 approach	fit with existing access
ANP	open agent networking	published project	required trust model
ERC-8004	registries and evidence	draft	value of blockchain use

Primary sources: [S28–S53, S62, S63, S67–S69, S94–S98, S108]. Responsibilities overlap; identity and authorisation cut across several functions.

Other connections worth following

Further models for agent networking are developing alongside these approaches. ANP investigates open agent communication. The ERC-8004 draft addresses blockchain-based registries for agent identity, reputation and validation. Such approaches become relevant to a business when they help it find or assess suitable partners. [S68, S69]

The sources in the compact protocol passport provide a way to follow these developments. Your own planning can stay close to the purchase: what should customers be able to delegate, where do they want to participate, and which connection brings the two together?

APPENDIX C

Five working templates for leaders

The one-page investment proposal

Customer task. Which need will be met better? **Baseline.** How often does it occur, how is it handled today and where do cost or errors arise? **Change.** What will the system do and what remains with people? **Dependencies.** Which data, rights, providers and agreements are required?

Evidence. Which complete process will actually be verified? **Economics.** What additional effect is expected and what migration might occur? **Expansion.** Under which observed conditions will the company expand, repair or stop? Do not manufacture precision. One decisive unknown is more useful than three unsupported decimal places.

The mandate card

Write the permitted action in one sentence. Identify principal, executing actor, object group, amount limit, currency, period and exceptions. State which changes require renewed approval. Identify where policy is enforced and how revocation operates.

An explicitly illustrative example: an agent may reorder approved consumables for one cost centre if total and delivery remain within an authorised range. New products and addresses require separate approval. Actual design depends on the organisation.

The outcome record

Record input, relevant configuration, data used and observed effect. Separate proposal, approved action and persisted result. Note a counterexample and the observed response. Evidence should be understandable without exposing every internal conversation.

When problems arise, this helps distinguish a missing capability, broken integration and a disproved assumption. That distinction saves more time than the general label “unreliable agent.”

The incident discussion

What did the customer want? What was permitted? What did the system do? Where did the first divergence occur? What consequences followed? How was state corrected? Which specific change prevents recurrence? Each answer needs an authoritative source.

Start with the first wrong information or action. Improving only the final wording is insufficient if the stored order is wrong. A more polite apology is good service, but not yet a technical repair.

The monthly leadership sheet

Show task volume, correctly completed processes, unauthorised executions, correction work and incremental economic contribution. Add the most important unresolved assumptions and the next concrete repair. Explain declining aggregate measures through individual causes.

Do not demand ever-longer reports. The sheet should enable a decision: continue, remove a constraint or change scope. If it merely records activity, it has missed its purpose.

APPENDIX D

Eight practical implementation patterns

Replenishment

Goal: reduce work for known products. **Prerequisites:** stable identity, quantities, price and delivery boundaries. **Start:** proposal with approval. **Test:** price change, unavailable quantity and unauthorised address. **Outcome:** correctly delivered repeat orders at acceptable total cost.

Technical product selection

Goal: improve suitability and compatibility. **Prerequisites:** confirmed attributes and explicit uncertainty. **Start:** search with a reasoned selection. **Test:** similarly named but unapproved alternative. **Outcome:** fewer wrong decisions and traceable technical approval.

Offer preparation

Goal: prepare complete offers faster. **Prerequisites:** buyer context, pricing rules and deliverability. **Start:** nonbinding draft. **Test:** quantity tiers, expiring terms and extra charges. **Outcome:** faster processing with correct conditions and fewer questions.

Service and exchange

Goal: resolve requests without repeated entry. **Prerequisites:** unique order and separate read and write permissions. **Start:** status and proposal. **Test:** expired window, changed surcharge and interrupted session. **Outcome:** correct resolution with understandable consent.

Catalogue quality

Goal: close decision-critical information gaps. **Prerequisites:** reliable sources and a business owner. **Start:** issue list and proposed corrections. **Test:** conflicting sources and unknown attributes. **Outcome:** fewer incorrect recommendations and less rework.

Merchant negotiation

Goal: develop permitted offers efficiently. **Prerequisites:** fixed commercial limits and binding offer records. **Start:** tightly bounded negotiation. **Test:** repeated discount pressure and changes beyond authority. **Outcome:** viable transactions rather than response speed alone.

Experience-assisted decisions

Goal: reduce spatial or aesthetic uncertainty. **Prerequisites:** fixed product truth and an accessible standard route. **Start:** one decision-relevant representation. **Test:** viewer failure and misleading visual implications. **Outcome:** improved decisions assessed against full cost.

Paid digital services

Goal: obtain external data or functions as needed. **Prerequisites:** quality, permitted payments and task budgets. **Start:** limited allowance. **Test:** retries, missing delivery and rising cost. **Outcome:** useful additional capability with traceable total expense.

What these patterns share

Each begins with a task, states prerequisites and ends with an observable result. Protocol selection follows. Reversing the order soon produces many interfaces and surprisingly few solved problems.

APPENDIX E

How the published theses developed

A connected argument over several years

The reading map connects earlier publications with this book’s themes. The technical chapters and protocol passport specify their respective document snapshots.

Thread	Publications	Contribution
Technology and creativity	Yin and Yang; early shopping and multimodal articles	human value and experience
Operational relief	January 2025 whitepaper; Agentic eCommerce	merchant work and complexity
From clicks to intent	Cheat Sheet; Thought-Chain; Customers Turn Into Code	delegation and journeys
Discovery and distribution	Software Ate the World; Discovery Layer	changing selection
Differentiation	Market Reset; adaptive storefronts	product value and commodity trap
Open AI and development	140,000 Lines; When Code Writes Code; Commerce Benchmark; Open-Weight Inflection	Open weights, customization and evidence
Prototypes	ACP with PayPal; generated storefronts	boundaries and inspectable results
Shared architecture	Strategy Book; Developer Guide	protocols, trust and operations
Executable systems	B2B interview; Execution Layer	business truth and permissions
Procurement experiment	Procurement Swarm	bounded end-to-end case

Further early writing on AI shopping, multimodality, spatial commerce and virtual try-ons appears as historical reading in the bibliography. Articles on AI-assisted development and the commerce benchmark add an engineering perspective.

Historical promises and benchmark results are not carried forward as evidence of current product capability. [S71–S78]

The engineering strand in the publications

My March and April 2025 essays discuss AI-assisted development through project work. The June 2026 commerce benchmark and July's "The Open-Weight Inflection" add open models and local execution. They inform Part V's question: which forms of customization become economical and remain maintainable? Project accounts and research on development productivity provide different kinds of evidence. [S76, S77, S78, S91, S118]

A reading map of the public experiments

The experiment boxes describe the task, setup, finding and status. References lead to the repositories and versions used. This map supports selective reading. [S99–S116]

Project group	Topic	Chapter
ATELIER / YOU and Storyfront	Personal presentation and product stories	6
AXP	Experience within the assistant	6
WebMCP and SwagA2UI	Tools and interfaces in the customer channel	9
Sales Agent Harness	Shop functions for several access routes	10
Merchant Quote Agent and rfq-bench	Negotiation within agreed boundaries	17
Shopware × Claude and Starter Kit	Proposals, approval and execution	11
x402	Paid digital services	8
Agent in a Box	Development environments for each project	13 and 16
SwagUcp and official plugin	Maintained protocol integration	19
Nyx and VEYNDO	Personal experiments with knowledge and adaptation	11 and 15

“After Turing” explores agent identification through behaviour or metadata. Identifying a request and authorising a purchase are different tasks. [S105]

Thinking ahead. Building ever since.

The path to this book begins with a question we have explored at Shopware for years: how do AI, commerce and experience fit together? Early ideas become articles, a whitepaper and, eventually, experiments with complete purchasing journeys.

June 2023 · Agents buy. People experience. In two articles, I describe AI agents for routine purchases, personal buying assistants and individually composed shopping experiences. I already see these as connected developments: agents take work off our hands, while people choose to spend time with an offer that interests them. [S169, S170]

21 December 2024 · The shop remains the foundation. Agents coordinate tasks through APIs; specialist commerce systems run the business behind them. That division of work becomes a central architectural question for agentic commerce. [S16]

17 January 2025 · Agentic eCommerce. In “Agentic eCommerce: Unlocking Next-Generation Differentiation”, I connect specialist agents, self-optimising stores and new opportunities to stand out in the market. [S15]

21 January 2025 · The first whitepaper. The dated original version develops these ideas in depth: AI autonomy, merchant operations and competition become a shared strategic challenge. [S01]

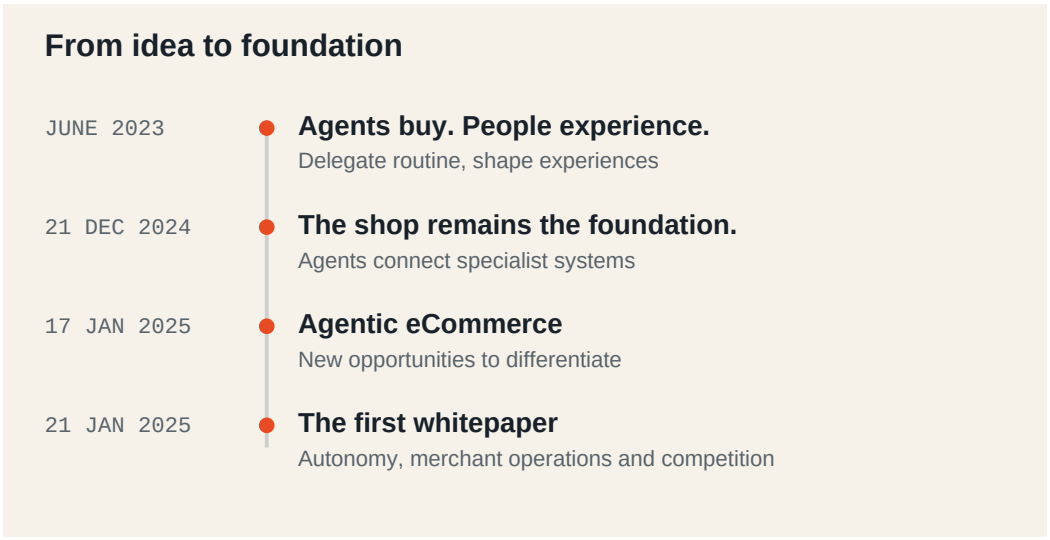


Figure 39. The early publications connect routine purchases and experience with an architecture and a broader strategic view. Sources S169, S170, S16, S15 and S01.

Putting the ideas to work.

1 October 2025 · Shopware, ACP and PayPal. I present our ACP implementation for Shopware and a working PayPal prototype. We can now test an agent-driven purchasing journey in practice: selection and conversation meet ordering and payment. [S20]

2026 · Work in the Shopware Agentic Commerce Lab. We develop and test shopping agents, open protocols and new merchant tools. Ideas become reference implementations, prototypes and insights we share with others. Chapter 17 explains how we approach that work. [S116]

October 2026 · From idea to book. The early theses, current developments and our Lab work come together in this handbook: 21 chapters on commerce strategy, technology and leadership, available in full in German and English, free to read online and as a PDF.

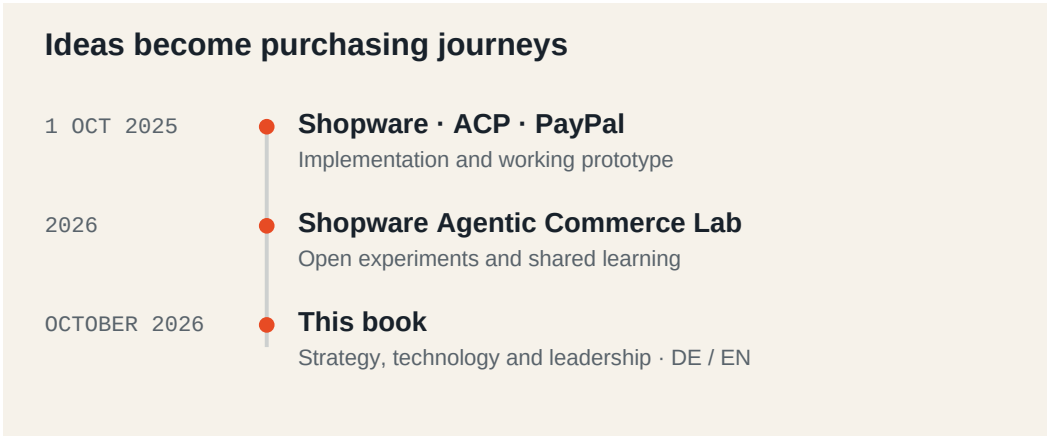


Figure 40. Practical work connects implementations, open Lab experiments and the handbook. Sources S20 and S116.

The market makes this division of work tangible, too. In April 2025, Amazon introduces Buy for Me as a limited beta for US customers: after their confirmation, an agent buys on a merchant’s website. With ACP, introduced in September 2025, the agent connects buyer and merchant; the merchant handles the order and fulfilment. To me, this supports the strategic direction: agents become a new route into commerce, and a capable commerce core becomes more valuable. [S171, S25]

How to read this book

The chapters connect my strategic arguments with published developments, personal experiments and practical decisions. “From the Lab” identifies projects from Shopware’s Agentic Commerce Lab; “From my experimental work” identifies my personal projects. The boxes describe the question, what we built, its main lesson and the next step. They provide a way to apply the ideas to your business and explore the code in more depth.

Nordstern, Werkpunkt and Lichtfeld are fictional merchants. They help us explore outdoor advice, spare-parts procurement and room design. The two anonymous real-world cases in Chapter 16 come from my merchant network. Model calculations state their assumptions; quotations and published facts refer to the sources in the appendix. That is also where you will find protocol versions.

Image credits for this edition

The 21 chapter vignettes form one conceptual visual story with the traveller, suitcase and compass agent. They are illustrations rather than captures of real projects. The Chapter 11 image moves to the merchant’s side: the merchant sets objectives and boundaries for the shop operator.

The author photograph and historical publication covers come from my website. Twelve original screenshots show the public projects discussed, Shopware documentation, and the Storyfront, Agentic Commerce Alliance and Readiness Scanner websites. Captions identify their provenance and, where documented, the capture date or project version. The team photograph comes from Shopware’s official team page. Diagrams are original explanatory models. Numerical graphics show the explicitly labelled calculations or dated published findings.

APPENDIX F

Sources and further reading

[S01] Stefan Hamann (2025-01-21). Agentic Commerce: How AI Autonomy Will Redefine Ecommerce Competition.

www.agentic-commerce.sh/whitepaper/Agentic%20Commerce%20-%20Full%20Whitepaper.pdf

[S02] Stefan Hamann (2025-05-16). Agentic Commerce Cheat Sheet for Ecom Leaders.

www.agentic-commerce.sh/whitepaper/Agentic%20Commerce%20-%20Cheat%20Sheet%201.0.pdf

[S03] Stefan Hamann (2026-01-20). Agentic Commerce Strategiebuch: Wie der Handel 2030 überlebt, v1.3.

www.agentic-commerce.sh/whitepaper/Agentic_Commerce_Strategiebuch_DE_v1.3_2026_1768888246873.pdf

[S04] Stefan Hamann (2026-01-20). Agentic Commerce Strategy Book: How Commerce Survives 2030, v1.3.

www.agentic-commerce.sh/whitepaper/Agentic_Commerce_Strategy_Book_EN_v1.3_2026_1768888246872.pdf

[S05] Stefan Hamann (2026-01-14). Agentic Commerce: Protocols, Standards and Reference Architectures.

www.agentic-commerce.dev/Agentic_Commerce_For_Developers.pdf

[S06] Stefan Hamann (2026-01-20). Commerce Is Quietly Changing Hands.

www.linkedin.com/pulse/commerce-quietly-changing-hands-stefan-hamann-uzclf/

[S07] Stefan Hamann (2026-03-12). Agentic Commerce Is Not Dead. The Discovery Layer Just Moved.

www.linkedin.com/pulse/agentic-commerce-dead-discovery-layer-just-moved-stefan-hamann-zzmmf/

[S08] Stefan Hamann (2026-04-30). I Stopped Building Pages and Started Generating Worlds. Anatomy of the Storefront That Comes After Yours.

www.linkedin.com/pulse/i-stopped-building-pages-started-generating-worlds-anatomy-hamann-qjere/

[S09] Stefan Hamann (2025-07-01). Frontends That Breathe: A Glimpse into Intent-Aware Commerce.

www.linkedin.com/pulse/frontends-breathe-glimpse-intent-aware-commerce-stefan-hamann-p2yce/

[S10] Stefan Hamann (2023-08-22). The Yin and Yang of Commerce: When Logic Met Artistry.

www.linkedin.com/pulse/yin-yang-commerce-when-logic-met-artistry-stefan-hamann/

[S11] Stefan Hamann (2025-05-14). When Customers Turn Into Code — The Playbook for Machine to Machine Commerce.

www.linkedin.com/pulse/when-customers-turn-code-playbook-machine-commerce-stefan-hamann-c7jue/

[S12] Stefan Hamann (2025-08-25). Agentic Commerce: A Market Reset Beyond Copycat Strategies.

www.linkedin.com/pulse/agentic-commerce-market-reset-beyond-copycat-stefan-hamann-ast5e/

[S13] Stefan Hamann (2025-05-09). Good-bye Click-Path, Hello Thought-Chain: Unlocking the Upside of Agentic Commerce 2025.

www.linkedin.com/pulse/good-bye-click-path-hello-thought-chain-unlocking-upside-hamann-xyh8e/

[S14] Stefan Hamann (2025-06-02). SOFTWARE ATE THE WORLD—NOW AGENTS ARE DEVOURING E-COMMERCE.

www.linkedin.com/pulse/software-ate-worldnow-agents-devouring-e-commerce-stefan-hamann-drg6e/

[S15] Stefan Hamann (2025-01-17). Agentic eCommerce: Unlocking Next-Generation Differentiation.

www.linkedin.com/pulse/agentic-ecommerce-unlocking-next-generation-stefan-hamann-qj1te/

[S16] Stefan Hamann (2024-12-21). AI Agents, APIs, and the Great eCommerce Disruption: Why We'll Never Have to Do "Manual Updates" Again.

www.linkedin.com/pulse/ai-agents-apis-great-ecommerce-disruption-why-well-never-hamann-tk3je/

[S17] Stefan Hamann (2025-05-02). AI Is Eating E-Commerce for Breakfast—Agentic Commerce Just Took the First Bite.

www.linkedin.com/pulse/ai-eating-e-commerce-breakfastagentic-commerce-just-took-hamann-bifie/

[S18] Stefan Hamann (2025-05-26). Blink, and It's Bought: AI Agents Ignite the Age of Agentic Commerce.

www.linkedin.com/pulse/blink-its-bought-ai-agents-ignite-age-agentic-commerce-stefan-hamann-dbqge/

[S19] Stefan Hamann (2025-07-17). When AI Meets Reality: What Anthropic's Project Vend Teaches Us About the Future of Agentic Commerce.

www.linkedin.com/pulse/when-ai-meets-reality-what-anthropics-project-vend-teaches-hamann-mhvme/

[S20] Stefan Hamann (2025-10-01). Agentic Commerce, Now Real: Shopware Launches ACP with a Live PayPal Prototype.

www.linkedin.com/pulse/agentic-commerce-now-real-shopware-launches-acp-live-paypal-hamann-rkrne/

[S21] Stefan Hamann (2025-11-03). Agentic Commerce 2.0: The Next Era of Commerce Has Begun.

www.linkedin.com/pulse/agentic-commerce-20-next-era-has-begun-stefan-hamann-agnhe/

[S22] Stefan Hamann (2026-09-03). „Agentic Commerce kann im B2B nur funktionieren, wenn die beteiligten Systeme eng zusammenspielen.".

www.ifhkoeln.de/agentic-commerce-kann-im-b2b-nur-funktionieren-wenn-die-beteiligten-systeme-eng-zusammenspielen/

[S23] Stefan Hamann (2026-09-07). What if a team of AI agents could help with B2B procurement?

www.linkedin.com/pulse/what-team-ai-agents-could-help-b2b-procurement-stefan-hamann-xrxee

[S24] Stefan Hamann (2026-09-04). Agents are becoming a buying surface. The shop underneath has to become an execution layer.

www.linkedin.com/pulse/agents-becoming-buying-surface-shop-underneath-has-become-hamann-4xwoe

[S25] OpenAI (2025-09-29). Buy it in ChatGPT: Instant Checkout and the Agentic Commerce Protocol.

openai.com/index/buy-it-in-chatgpt/

[S26] OpenAI (2026-03-24). Powering Product Discovery in ChatGPT.

openai.com/index/powering-product-discovery-in-chatgpt/

[S27] Google Vidhya Srinivasan (2026-01-11). New tech and tools for retailers to succeed in an agentic shopping era.

blog.google/products/ads-commerce/agentic-commerce-ai-tools-protocol-retailers-platforms/

[S28] Universal Commerce Protocol contributors (n.d.). Universal Commerce Protocol documentation.
Document snapshot: Release v2026-08-25; current documentation checked separately.

ucp.dev/

[S29] OpenAI and Stripe (n.d.). Agentic Commerce Protocol.
Document snapshot: Beta; dated specification 2026-04-17.

www.agenticcommerce.dev/

[S30] Model Context Protocol contributors (2026-07-28). Model Context Protocol specification.
Document snapshot: 2026-07-28.

modelcontextprotocol.io/specification/2026-07-28

[S31] Model Context Protocol contributors (2026-07-28). Tools.
Document snapshot: 2026-07-28.

modelcontextprotocol.io/specification/2026-07-28/server/tools

[S32] A2A Project (2026). Agent2Agent protocol specification v1.0.0.
Document snapshot: 1.0.0.

a2a-protocol.org/v1.0.0/specification/

[S33] Sierra (2026-10-06). Introducing Personal Agent Protocol.

sierra.ai/blog/introducing-personal-agent-protocol

[S34] Personal Agent Protocol authors (n.d.). Personal Agent Protocol specification Draft 0.1.
Document snapshot: Draft 0.1; retrieved 15 October 2026; no publication or modification date stated.

personalagentprotocol.org/docs/spec

[S35] Personal Agent Protocol authors (n.d.). Operations extension version 1 for draft 0.1.
Document snapshot: Operations v1 for Draft 0.1; retrieved 15 October 2026.

personalagentprotocol.org/docs/extensions/operations

[S36] Personal Agent Protocol authors (n.d.). Path to 1.0.
Document snapshot: Roadmap to 1.0; retrieved 15 October 2026.

personalagentprotocol.org/docs/path-to-1

[S37] Personal Agent Protocol authors (2026-10-14). Guides.
Document snapshot: Guides checked 2026-10-14.

personalagentprotocol.org/docs/guides

[S38] Google AP2 contributors (n.d.). Agent Payments Protocol documentation v0.2.
Document snapshot: v0.2.

ap2-protocol.org/

[S39] Nishant Kaushik FIDO Alliance (2026-05-26). Building the Trust Layer for Agentic Payments with AP2 and Verifiable Intent.

fidoalliance.org/building-the-trust-layer-for-agentic-payments-with-ap2-and-verifiable-intent/

[S40] Visa (n.d.). Getting started with Trusted Agent Protocol.

developer.visa.com/docs/trusted-agent-protocol

[S41] Mastercard Sophie Hares (2026-10-08). An age-old question for the AI era: Can I see some ID?

www.mastercard.com/us/en/news-and-trends/stories/2026/agentic-commerce-identity-know-your-agent.html

[S42] Stripe (2026-03-03). Supporting additional payment methods for agentic commerce.

stripe.com/blog/supporting-additional-payment-methods-for-agentic-commerce

[S43] Stripe (2026-04-29). Giving agents the ability to pay.

stripe.com/blog/giving-agents-the-ability-to-pay

[S44] Stripe and Tempo (2026-03). Introducing the Machine Payments Protocol.

stripe.com/blog/machine-payments-protocol

[S45] MPP contributors (n.d.). Machine Payments Protocol.

mpp.dev/

[S46] x402 Foundation contributors (n.d.). x402 protocol and reference implementation.

github.com/x402-foundation/x402

[S47] PayPal (2026-06-10). Agent Ready overview.

developer.paypal.com/agent-ready/overview/

[S48] PayPal (2026-09-16). WebMCP is here PayPal is helping every merchant become agent ready.

developer.paypal.com/community/blog/WebMCP_PayPal_Agent_Ready/

[S49] Chrome for Developers (2026-10-07). WebMCP.
Document snapshot: Proposed browser standard; documentation updated 2026-10-07.

developer.chrome.com/docs/ai/webmcp

[S50] Agentic Commerce Lab (n.d.). AXP — Agentic Experience Protocol.
Document snapshot: Research prototype 2026-01-13.

github.com/agentic-commerce-lab/AXP-protocol

[S51] AG UI contributors (n.d.). AG-UI Overview: The Agent–User Interaction Protocol.

docs.ag-ui.com/introduction

[S52] Agentic Resource Discovery contributors (n.d.). Agentic Resource Discovery Specification.

agenticresourcediscovery.org/spec/

[S53] Decagon Harry Gao (2026-10-06). Introducing the Personal Agent Consent & Trust Protocol (PACT).

decagon.ai/blog/introducing-the-personal-agent-consent-trust-protocol-pact

[S54] Agentic Commerce Alliance (n.d.). Mission governance and membership.
Document snapshot: Public mission, FAQ and governance principles; merchant membership and competing platforms checked on 15 October 2026.

www.agentic-commerce.org/

[S55] Agentic Commerce Alliance (n.d.). Agentic Commerce Maturity Index ACMI.
Document snapshot: 24 questions; Operations, Discovery, Transactions, Fulfillment; live browser page checked 15 October 2026.

acmi.agentic-commerce.org/

[S56] Agentic Commerce Alliance (n.d.). Agentic Readiness Scanner methodology.
Document snapshot: Live methodology: 32 checks in 8 categories, including two unweighted checks; read-only GET, no login or JavaScript; checked 15 October 2026.

scanner.agentic-commerce.org/methodology

[S57] Anthropic (2025-06-27). Project Vend: Can Claude run a small shop? (And why does that matter?).

www.anthropic.com/research/project-vend-1

[S58] Anthropic (2025-12-18). Project Vend: Phase two.

www.anthropic.com/research/project-vend-2

[S59] European Union (2016). General Data Protection Regulation, Regulation (EU) 2016/679.

eur-lex.europa.eu/eli/reg/2016/679/oj/eng

[S60] European Commission (n.d.). AI Act: regulatory framework and implementation.

digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

[S61] European Union (2011). Consumer Rights Directive 2011/83/EU.

eur-lex.europa.eu/eli/dir/2011/83/oj/eng

[S62] IETF (2023). RFC 9449: OAuth 2.0 Demonstrating Proof of Possession (DPoP).

www.rfc-editor.org/rfc/rfc9449

[S63] IETF (2025). RFC 9727: api-catalog: A Well-Known URI and Link Relation to Help Discovery of APIs.

www.rfc-editor.org/rfc/rfc9727

[S64] Microsoft (2026-01-08). Microsoft propels retail forward with agentic AI capabilities that power intelligent automation for every retail function.

news.microsoft.com/source/2026/01/08/microsoft-propels-retail-forward-with-agentic-ai-capabilities-that-power-intelligent-automation-for-every-retail-function/

[S65] Google (2026-05-20). Google Marketing Live 2026.

blog.google/products/ads-commerce/google-marketing-live-2026-collection/

[S66] Stefan Hamann (n.d.). Agentic Commerce publication index and thesis.

www.agentic-commerce.sh/

[S67] MCP Apps contributors (n.d.). MCP Apps documentation.

apps.extensions.modelcontextprotocol.io/api/

[S68] ANP contributors (n.d.). Agent Network Protocol.

agent-network-protocol.com/

[S69] ERC 8004 authors (n.d.). ERC-8004: Trustless Agents.
Document snapshot: Draft.

eips.ethereum.org/EIPS/eip-8004

[S70] OWASP Foundation (n.d.). OWASP Top 10 for Large Language Model Applications.

owasp.org/projects/top-10-for-large-language-model-applications

[S71] Stefan Hamann (2023-07-03). Clippy’s Comeback: Navigating the Maze of E-commerce with Microsoft’s New AI-Powered Shopping Tools.

www.linkedin.com/pulse/clippys-comeback-navigating-maze-e-commerce-new-shopping-hamann/

[S72] Stefan Hamann (2023-11-20). Riding the AI Tsunami: A Commerce Odyssey in the Age of Machine Minds.

www.linkedin.com/pulse/riding-ai-tsunami-commerce-odyssey-age-machine-minds-stefan-hamann-o20ze/

[S73] Stefan Hamann (2023-09-26). The Dawn of a New Multimodal Era: OpenAI’s ChatGPT Levels Up and What It Means for Commerce.

www.linkedin.com/pulse/dawn-new-multimodal-era-openais-chatgpt-levels-up-what-stefan-hamann/

[S74] Stefan Hamann (2024-01-29). Shopware’s Strategic Vision in Spatial Commerce and AI Integration.

www.linkedin.com/pulse/shopwares-strategic-vision-spatial-commerce-ai-stefan-hamann-pkowe/

[S75] Stefan Hamann (2025-01-02). AI and Virtual Try-Ons in Fashion: A Deeper Dive into the Tech That’s Changing eCommerce.

www.linkedin.com/pulse/ai-virtual-try-ons-fashion-deeper-dive-tech-thats-changing-hamann-okqie/

[S76] Stefan Hamann (2026-06-29). Agentic Commerce Benchmark: Open Source Just Pulled Level with the Frontier.

www.linkedin.com/pulse/agentic-commerce-benchmark-open-source-just-pulled-level-hamann-wa52e/

[S77] Stefan Hamann (2025-04-29). When Code Writes Code: A Quiet Revolution in eCommerce Engineering.

www.linkedin.com/pulse/when-code-writes-quiet-revolution-ecommerce-stefan-hamann-poqzc/

[S78] Stefan Hamann (2025-03-30). How AI-Powered Vibe Coding Helped Me Write 140,000 Lines of Deployable Code — and What It Means for Engineering.

www.linkedin.com/pulse/how-ai-powered-vibe-coding-helped-me-write-140000-lines-stefan-hamann-fez3c/

[S79] McKinsey & Company (2025-10-17). The agentic commerce opportunity: How AI agents are ushering in a new era for consumers and merchants.

www.mckinsey.com/capabilities/quantumblack/our-insights/the-agentic-commerce-opportunity-how-ai-agents-are-ushering-in-a-new-era-for-consumers-and-merchants

[S80] Bain & Company (2025-12-17). 2030 Forecast: How Agentic AI Will Reshape US Retail.

www.bain.com/insights/2030-forecast-how-agentic-ai-will-reshape-us-retail-snap-chart/

[S81] Gartner (2025-10-21). Gartner Unveils Top Predictions for IT Organizations and Users in 2026 and Beyond.

www.gartner.com/en/newsroom/press-releases/2025-10-21-gartner-unveils-top-predictions-for-it-organizations-and-users-in-2026-and-beyond

[S82] Gartner (2026-01-15). Gartner Predicts 60% of Brands Will Use Agentic AI to Deliver Streamlined One-to-One Interactions by 2028.

www.gartner.com/en/newsroom/press-releases/2026-01-15-gartner-predicts-60-percent-of-brands-will-use-agentic-ai-to-deliver-streamlined-one-to-one-interactions-by-2028

[S83] Gartner (2025-06-25). Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027.

www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027

[S84] McKinsey & Company (2026-03-11). McKinsey: KI-Helfer im Einkaufsalltag angekommen.

www.mckinsey.de/news/presse/2026-03-11-agentic-commerce-eu-survey

[S85] Adobe Digital Insights (2026-09-25). UK Online Spending Trends: January–August 2026.

business.adobe.com/uk/blog/uk-digital-insights-trends-august-2026

[S86] Open Source Initiative (2024). The Open Source AI Definition 1.0.

opensource.org/ai/open-source-ai-definition

[S87] METR (2025-07-10). Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity.

metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/

[S88] METR (2026-02-24). We are Changing our Developer Productivity Experiment Design.

metr.org/blog/2026-02-24-uplift-update/

[S89] Shopware (2023-05-25). Shopware enthüllt bahnbrechende AI-Innovationen beim SCD: „AI Copilot“ veröffentlicht.

www.shopware.com/de/presse/pressemitteilungen/shopware-enthuellt-bahnbrechende-ai-innovationen-beim-scd-ai-copilot/

[S90] Shopware (2024-01-25). Stefan Hamann on ecommerce evolution and AI's role in 2024 and beyond.

www.shopware.com/en/news/stefan-hamann-on-ecommerce-evolution-and-ais-role-in-2024-and-beyond/

[S91] Stefan Hamann (2026-07-07). The Open-Weight Inflection.

www.linkedin.com/pulse/open-weight-inflection-stefan-hamann-eez1e

[S92] Mozilla (2026-09). The State of Open Source AI, v1.1.
Document snapshot: v1.1, September 2026.

stateofopensource.ai/

[S93] GitHub (2025-10-28). Octoverse: A new developer joins GitHub every second as AI leads TypeScript to #1.
Document snapshot: Octoverse 2025; page updated 2026-02-28.

github.blog/news-insights/octoverse/octoverse-a-new-developer-joins-github-every-second-as-ai-leads-typescript-to-1/

[S94] Universal Commerce Protocol (n.d.). Versioning.

ucp.dev/versioning/

[S95] Universal Commerce Protocol (2026-08-25). Release v2026-08-25.

github.com/Universal-Commerce-Protocol/ucp/releases/tag/v2026-08-25

[S96] Universal Commerce Protocol (n.d.). Specification overview.

ucp.dev/specification/overview/

[S97] Agentic Commerce Protocol (n.d.). Documentation.

www.agenticcommerce.dev/docs

[S98] Agentic Commerce Protocol (2026-04-17). Specification repository: dated release 2026-04-17 and development branches.
Document snapshot: Beta; 2026-04-17; unreleased development kept separate.

github.com/agentic-commerce-protocol/agentic-commerce-protocol

[S99] Agentic Commerce Lab / Shopware (n.d.). rfq-bench: public repository and documentation.
Document snapshot: e95cf6dbd58b.

github.com/agentic-commerce-lab/rfq-bench/tree/e95cf6dbd58b5e6584c6389ab8b21c9009003e19

[S100] Agentic Commerce Lab / Shopware (n.d.). merchant-quote-agent-plugin: public repository and documentation.
Document snapshot: 73f8b4bb0e93.

github.com/agentic-commerce-lab/merchant-quote-agent-plugin/tree/73f8b4bb0e93f13cc4b9f10696c8e88eeab39119

[S101] Agentic Commerce Lab / Shopware (n.d.). shopware-claude-commerce: public repository and documentation.
Document snapshot: e0a3e446992c.

github.com/agentic-commerce-lab/shopware-claude-commerce/tree/e0a3e446992c071a34896f5c25d6e6e5b5346dd0

[S102] Agentic Commerce Lab / Shopware (n.d.). shopping-assistant-starter-kit: public repository and documentation.
Document snapshot: ee1c82d613bd.

github.com/agentic-commerce-lab/shopping-assistant-starter-kit/tree/ee1c82d613bd410929fa1cda7ac3ad67e6b7d50f

[S103] Agentic Commerce Lab / Shopware (n.d.). AXP-protocol: public repository and documentation.
Document snapshot: a5ed802da7e4.

github.com/agentic-commerce-lab/AXP-protocol/tree/a5ed802da7e41e1621d0265779e9a0a73214b8c9

[S104] Agentic Commerce Lab / Shopware (n.d.). webmcp-plugin: public repository and documentation.
Document snapshot: 6d76617e919b.

github.com/agentic-commerce-lab/webmcp-plugin/tree/6d76617e919b608921c5539b8bc3a782ad76f69d

[S105] Agentic Commerce Lab / Shopware (n.d.). agent-signals-research-notes: public repository and documentation.
Document snapshot: 355d0d9cc6d7.

github.com/agentic-commerce-lab/agent-signals-research-notes/tree/355d0d9cc6d7aa35f4ae8b1fc105efef7641ba30

[S106] Agentic Commerce Lab / Shopware (n.d.). sales-agent-harness: public repository and documentation.
Document snapshot: 51efcee7d768.

github.com/agentic-commerce-lab/sales-agent-harness/tree/51efcee7d7680f8e3fe8444a4dbed029129caf94

[S107] Agentic Commerce Lab / Shopware (n.d.). SwagUcp: public repository and documentation.
Document snapshot: ac864ede22c7; archived; reference only.

github.com/agentic-commerce-lab/SwagUcp/tree/ac864ede22c751fff7bab6b31c0b34be048ce929

[S108] Agentic Commerce Lab / Shopware (n.d.). SwagA2UI: public repository and documentation.
Document snapshot: a3d11b669a49.

github.com/agentic-commerce-lab/SwagA2UI/tree/a3d11b669a49a84a40e4256ca5671dbe874073c1

[S109] Agentic Commerce Lab / Shopware (n.d.). agent-in-a-box: public repository and documentation.
Document snapshot: 7ac21627dac9.

github.com/agentic-commerce-lab/agent-in-a-box/tree/7ac21627dac96c178fecb8771f9b525cb033a7e8

[S110] Agentic Commerce Lab / Shopware (n.d.). x402-payment-plugin: public repository and documentation.
Document snapshot: eb754b1a8b2c.

github.com/agentic-commerce-lab/x402-payment-plugin/tree/eb754b1a8b2cf87468cd2a9a82273d6c8eed1c7f

[S111] Stefan Hamann (n.d.). nyx-local-ai: public repository and documentation.
Document snapshot: 59957fffa06f.

github.com/sthamann/nyx-local-ai/tree/59957fffa06f3888840d915a809dd4d8c2dad2a2

[S112] Stefan Hamann (n.d.). veyndo: public repository and documentation.
Document snapshot: 74dbc1919ebf; Sustainable Use License 1.0 (source-available, not OSI open source).

github.com/sthamann/veyndo/tree/74dbc1919ebf1ca257261ca1a5102dba472b399a

[S113] Stefan Hamann (n.d.). atelier-you: public repository and documentation.
Document snapshot: 871ee3c6aba1.

github.com/sthamann/atelier-you/tree/871ee3c6aba1569cd39a6876daffdba14a0368a5

[S114] Stefan Hamann (n.d.). ACP-shopware: public repository and documentation.
Document snapshot: 5289cfa32fa2.

github.com/sthamann/ACP-shopware/tree/5289cfa32fa2f2034f1232ae2648b61ebf5b602e

[S115] Shopware (n.d.). Agentic Commerce: official Shopware plugin repository.
Document snapshot: 91a38853e32a.

github.com/shopware/agentic-commerce/tree/91a38853e32a52afde03de8065db438d5d2d3d45

[S116] Agentic Commerce Lab / Shopware (n.d.). The Agentic Commerce Lab: mission and public experiments.

github.com/agentic-commerce-lab

[S117] J. Yannis Bakos (1997-12). Reducing Buyer Search Costs: Implications for Electronic Marketplaces. Management Science 43(12), 1676–1692.

people.stern.nyu.edu/bakos/emkts.pdf

[S118] Mert Demirer, Leon Musolff and Liyuan Yang (2026-05). Writing Code vs. Shipping Code: Productivity Effects Across Generations of AI Coding Tools. NBER Working Paper 35275.
Document snapshot: Revised September 2026; working paper.

www.nber.org/papers/w35275

[S119] European Data Protection Board (2024-12-17). Opinion 28/2024 on certain data protection aspects related to the processing of personal data in the context of AI models.

www.edpb.europa.eu/system/files/2024-12/edpb_opinion_202428_ai-models_en.pdf

[S120] European Commission (n.d.). Consumer Rights Directive: current overview and guidance.

commission.europa.eu/law/law-topic/consumer-protection-law/consumer-contract-law/consumer-rights-directive_en

[S121] Shopware (n.d.). Shopping Experiences.

docs.shopware.com/en/shopware-6-en/content/ShoppingExperiences

[S122] Shopware (n.d.). Copilot: analysis, recommendations and controlled execution.
Document snapshot: Stable Copilot release; permissions, review and approval; feature scope depends on the Shopware plan.

www.shopware.com/en/products/shopware-intelligence/copilot/

[S123] Shopware (2026-06-10). Shopware Community Day 2026: Shopware Becomes the Open Commerce Infrastructure for the Agentic Era.

www.shopware.com/en/press/press-releases/shopware-community-day-2026-shopware-becomes-the-open-commerce-infrastructure-for-the-agentic-era/

[S124] Stefan Hamann (n.d.). VEYNDO: Cognitive commerce — connected implementation and remaining work.
Document snapshot: 74dbc1919ebf.

github.com/sthamann/veyndo/blob/74dbc1919ebf1ca257261ca1a5102dba472b399a/docs/cognitive-commerce.md

[S125] Stefan Hamann (n.d.). VEYNDO: Knowledge workspace.
Document snapshot: 74dbc1919ebf.

github.com/sthamann/veyndo/blob/74dbc1919ebf1ca257261ca1a5102dba472b399a/docs/knowledge-workspace.md

[S126] Stefan Hamann (n.d.). VEYNDO: Storyfront connector.
Document snapshot: 74dbc1919ebf.

github.com/sthamann/veyndo/blob/74dbc1919ebf1ca257261ca1a5102dba472b399a/docs/storyfront.md

[S127] Stefan Hamann (n.d.). VEYNDO: Trusted identity and hosted frontend integration.
Document snapshot: 74dbc1919ebf.

github.com/sthamann/veyndo/blob/74dbc1919ebf1ca257261ca1a5102dba472b399a/docs/experience-integration.md

[S128] Shopware (n.d.). The management of shopware AG.
Document snapshot: Management listing: Founder and Chief Agentic Officer.

www.shopware.com/en/company/management/

[S129] Stefan Hamann / VEYNDO × Storyfront (n.d.). Storyfront fashion experience: editorial product world and linked product detail.
Document snapshot: Live public demo, captured 14 October 2026; fictional test catalog.

pme-legends-83294f02d8.veyndo.ai/

[S130] Shopware (n.d.). Agentic Commerce Lab: team, mission and working model.

careers.shopware.com/de/team/agentic-commerce-lab/

[S131] Shopware (2026-10). Shopware Release News Oktober 2026: Das ist neu in 6.7.15.0.

www.shopware.com/de/news/shopware-6-release-news-oktober-2026/

[S132] Bundesrepublik Deutschland (n.d.). BGB § 164: Wirkung der Erklärung des Vertreters.

www.gesetze-im-internet.de/bgb/__164.html

[S133] Bundesrepublik Deutschland (n.d.). BGB § 177: Vertragsschluss durch Vertreter ohne Vertretungsmacht.

www.gesetze-im-internet.de/bgb/__177.html

[S134] Bundesrepublik Deutschland (n.d.). BGB § 312j: Besondere Pflichten im elektronischen Geschäftsverkehr gegenüber Verbrauchern.

www.gesetze-im-internet.de/bgb/__312j.html

[S135] Bundesrepublik Deutschland (n.d.). BGB § 355: Widerrufsrecht bei Verbraucherverträgen.

www.gesetze-im-internet.de/bgb/__355.html

[S136] Bundesrepublik Deutschland (n.d.). BGB § 312g: Widerrufsrecht.

www.gesetze-im-internet.de/bgb/__312g.html

[S137] European Banking Authority (2019-03-01). Single Rulebook Q&A 2018_4031: Applicability of SCA to card payments initiated by the payee only.

www.eba.europa.eu/single-rule-book-qa/qna/view/publicId/2018_4031

[S138] European Commission (n.d.). DMA designated gatekeepers and core platform services.

digital-markets-act.ec.europa.eu/gatekeepers-portal_en

[S139] European Commission (2023-07-21). Guidelines on horizontal co-operation agreements, section 6.2.2: algorithms and information exchange.

eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A52023XC0721%2801%29

[S140] Wero / EPI (n.d.). E/M-commerce: merchant payment flow.

wero-wallet.eu/be-nl/e-m-commerce

[S141] Shopify (2026-01-11). The agentic commerce platform: Shopify connects any merchant to every AI conversation.

www.shopify.com/news/ai-commerce-at-scale

[S142] Amazon (2026-03). Amazon is making it easier for merchants to sell from external stores.

www.aboutamazon.com/news/retail/amazon-shop-direct-external-stores

[S143] Anthropic (2024-11-25). Introducing the Model Context Protocol.

www.anthropic.com/news/model-context-protocol

[S144] Stefan Hamann (2026-10-15). Two qualitative merchant examples from the author's network.
Document snapshot: Unpublished author account; identifying details reduced for the 18 October 2026 edition.

Unpublished author account.

[S145] Shopware (2026-08-11). New Management Structure at shopware AG.

www.shopware.com/en/press/press-releases/new-management-structure-at-shopware-ag/

[S146] Oberlandesgericht Düsseldorf (2016-05-19). Urteil vom 19. Mai 2016, I-16 U 72/15: automatisierte Auftragsbestätigung und Vertragsschluss.
Document snapshot: Official judicial publication; paragraphs 56–69, particularly automatic acceptance and interpretation of electronic declarations.

nrwe.justiz.nrw.de/olgs/duesseldorf/j2016/I_16_U_72_15_Urteil_20160519.html

[S147] European Commission (n.d.). Data access for business users — Article 6(10) DMA.
Document snapshot: Article 6(10) business-user data access; provider API and request links checked.

digital-markets-act.ec.europa.eu/businesses-portal/data-access/data-access-business-users-article-610-dma_en

[S148] Bundesministerium der Finanzen (2025). GoBD — AO-Handbuch 2025, Anhang 33, insbesondere Rz. 142–144 und 151–155.

amtliche-handbuecher.bundesfinanzministerium.de/ao/2025/Anhaenge/BMF-Schreiben-und-gleichlautende-Laendererlasse/Anhang-33/anhang-33.html

[S149] Bundesministerium der Finanzen (2025-07-14). Zweite Änderung der GoBD.

www.bundesfinanzministerium.de/Content/DE/Downloads/BMF_Schreiben/Weitere_Steuerthemen/Abgabenordnung/2025-07-14-GoBD-2-aenderung.pdf?__blob=publicationFile&v=2

[S150] Shopware (n.d.). Shopware 6 — public repository, README and MIT licence.

github.com/shopware/shopware

[S151] Shopware (n.d.). Shopware CLI — local development, extension validation and agent skills.

github.com/shopware/shopware-cli

[S152] Shopware (2026-09). Shopware Release News September 2026: Experience Studio Research Preview.
Document snapshot: Experience Studio remains a Research Preview; product landing pages, hero video and Shopware export.

www.shopware.com/de/news/shopware-6-release-news-september-2026/

[S153] Shopware (n.d.). Agentic Commerce: official plugin and compatibility documentation.
Document snapshot: 86d4c8016dae; public extension, supported Shopware lines and UCP capabilities.

github.com/shopware/agentic-commerce/tree/86d4c8016daed45585b835fa9ceb2c143f223c28

[S154] OpenAI (n.d.). Overview of OpenAI Crawlers.
Document snapshot: Search, training and user-initiated access; independent settings.

developers.openai.com/api/docs/bots

[S155] Bundesrepublik Deutschland (n.d.). § 7 UWG: Unzumutbare Belästigungen.
Document snapshot: Section 7(2), point 2, and Section 7(3).

www.gesetze-im-internet.de/uwg_2004/__7.html

[S156] Bundesrepublik Deutschland (n.d.). § 87 BetrVG: Mitbestimmungsrechte.
Document snapshot: Section 87(1), point 6.

www.gesetze-im-internet.de/betrvg/__87.html

[S157] Bundesarbeitsgericht (2022-03-08). Beschluss vom 8. März 2022, 1 ABR 20/21.
Document snapshot: Paragraphs 29–31: objective suitability for monitoring performance or behaviour.

www.bundesarbeitsgericht.de/entscheidung/1-abr-20-21/

[S158] OpenAI (n.d.). API Pricing.
Document snapshot: Input, output, cached input and tool charges; no fixed model price quoted.

developers.openai.com/api/docs/pricing

[S159] Bundesrepublik Deutschland (n.d.). Artikel 246a § 1 EGBGB: Informationspflichten.
Document snapshot: Article 246a section 1(1), point 6: personalized price based on automated decision-making.

www.gesetze-im-internet.de/bgbeg/art_246a__1.html

[S160] European Union (2019-11-27). Directive (EU) 2019/2161: consumer protection modernization.
Document snapshot: Recital 45: personalized pricing versus non-personalized dynamic pricing.

eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019L2161

[S161] European Commission / AI Act Service Desk (n.d.). Article 50: Transparency obligations for providers and deployers of certain AI systems.
Document snapshot: Article 50(1) and (5); consolidated text as at 27 July 2026; application from 2 August 2026 checked against Commission guidance.

ai-act-service-desk.ec.europa.eu/en/ai-act/article-50

[S162] European Commission / AI Act Service Desk (n.d.). Article 3: Definitions — provider and deployer.
Document snapshot: Article 3(3) and (4); consolidated text as at 27 July 2026.

ai-act-service-desk.ec.europa.eu/en/ai-act/article-3

[S163] Google / Suresh Ganapathy (2026-10-15). An intelligent new shopping cart.
Document snapshot: US launch across Search and Gemini; merchant cart handover and checkout; merchant remains seller.

blog.google/products-and-platforms/products/shopping/universal-cart-launch/

[S164] Google for Developers (2026-10-16). Cart API — Google UCP Guide.
Document snapshot: Cart capability from UCP 2026-04-08; Google CreateCart handover subset; no synchronization or native-checkout requirement.

developers.google.com/merchant/ucp/guides/overview/cart-overview

[S165] Stripe (2026-09-08). Stripe helps Muse, Meta's new personal AI agent, shop across the internet with Link.
Document snapshot: US offering; per-purchase total approval; saved payment methods and purchase-scoped virtual cards.

stripe.com/gb/newsroom/news/stripe-helps-meta-muse-shop-with-link

[S166] Meta (2026-09-29). The Future Is for Everyone: Muse for Small Business.
Document snapshot: Announcement updated 2026-10-08; connected business tools; approval before publishing, sending or spending.

about.fb.com/news/2026/09/introducing-muse-small-business/

[S167] OpenAI (2026-09-16). Reimagining advertising with AI.
Document snapshot: Sponsored Agents announced as a test; business-sponsored conversation after an ad click.

openai.com/index/reimagining-advertising-with-ai/

[S168] OpenAI Developers (n.d.). Products — Agentic Commerce.
Document snapshot: Stable product feed documentation accessed 2026-10-18; discovery, Ads and separately enabled checkout distinguished.

developers.openai.com/commerce/specs/file-upload/products

[S169] Stefan Hamann (2023-06-14). Riding the AI Wave: Don't Be the One Who Thought the Internet Was Just a Fad.

stefan-hamann.de/artikel/riding-the-ai-wave-don-t-be-the-one-who-thought-the-internet-was-just-a-fad-2023/

[S170] Stefan Hamann (2023-06-15). Beyond Personas: The New Era of Hyper-Personalization in eCommerce.

stefan-hamann.de/artikel/beyond-personas-the-new-era-of-hyper-personalization-in-ecommerce-2023/

[S171] Amazon Staff (2025-04-03). Amazon's new 'Buy for Me' feature helps customers find and buy products from other brands' sites.

www.aboutamazon.com/news/retail/amazon-shopping-app-buy-for-me-brands