CARVE
CONSULTING

Purchasing Channel Management
A framework we use with clients

The purchasing channel most CFOs never designed

By Jesper Sten Madsen

August 2026

Key takeaways

Purchasing is not a single process, but a portfolio of channels shaped by the type and characteristics of each purchase, as well as the operational system that contains the core information about what is being bought. Buying goods, raw materials and services therefore require different approaches, starting with a buy/pay flow.

The contract must remain the source of truth, yet experience show that many organizations still rely on manual work to connect contract terms with transactions - or fail to make that connection at all. This creates exceptions, spend leakage and unnecessary operational cost.

AI introduces opportunities that were previously out of reach. AI agents can act as a connective layer across contracts, maintenance systems, P2P platforms and other legacy tools to validate commitments and invoices against contract terms without requiring every system to be integrated.

This article explores how well-defined purchasing channel governance, combined with AI, can reduce cost leakage, and create a more resilient P2P execution model.

The purchasing channel most CFOs never designed

Most finance leaders can tell you what percentage of spend runs through a purchase order. Fewer can tell you why a given euro of spend ended up in that flow at all, or whether the control wrapped around it is enforceable by the system that originated the transaction. That gap is where P2P value quietly leaks away, long after the contract is signed and long before anyone notices.

The instinct in most transformation programs is to design purchasing controls from the top down: pick a small number of approval flows, three-way match, maybe a card program for the small stuff, and push every vendor and every transaction through them. It reads well in a steering committee deck. It rarely survives contact with how the organization buys in reality.

Purchasing is not one process. It is a portfolio of channels

A purchase order for pallets of raw material, a facilities contractor billing against a maintenance milestone, and a SaaS subscription renewing quietly every January are not variations on the same process. They are different types of purchases with different characteristics:

Physical goods
That can be received against a quantity and price.

Services
Delivered against time and a milestone.

Subscriptions
That renew against a contract with no receipt at all.

Each of those characteristics points to a different buy/pay flow: three-way match for goods ordered through a catalogue, evaluated receipt settlement where the goods receipt itself is enough to trigger payment, service validation tied to milestones for contracted services, recurring invoicing against an approved contract for subscriptions, and a residual invoice-only channel for the purchase order numbers that never existed in the first place.

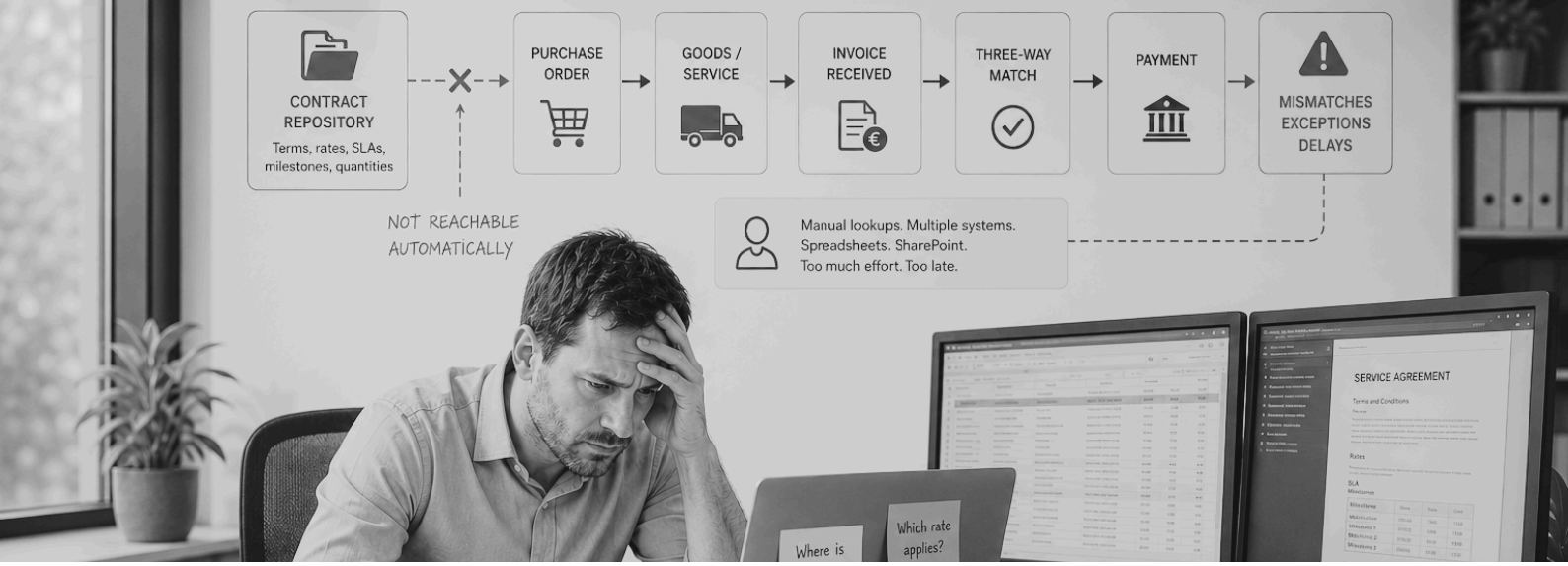
Treated as a menu, this is a useful start. Treated as the whole answer, it still misses the piece that determines whether any of it works.

The control is only as real as the system behind it

Here is the part that gets skipped: every one of those purchase types is captured, in practice, by a specific operational system, and that system's capabilities set a hard ceiling on which control is even possible. Goods purchases run through a P2P platform that supports catalogues and goods receipts. Services run through whatever system manages the work, a maintenance system, a project management system, a building management system, and that system defines how (or whether) a milestone or completion event can be tracked and linked to an invoice. Subscriptions live in IT asset registers, maintenance platforms, or wherever the contract itself is administered.

Design a three-way match control for a purchase type whose originating system has no concept of a goods receipt, and you have not created control, you have created a manual workaround, or worse, a control that gets silently bypassed. Design an assumed-receipt flow for a services category where the operating system already tracks milestone completion in detail, and you have thrown away perfectly good evidence and replaced it with trust. Either way, the mismatch between purchase type, operational system, and buy/pay flow is exactly where the "leak" in the P2P chain opens up: exceptions pile up, invoices get parked, people get pulled in to manually reconcile things the process should have handled on its own.

This is the sequence that actually holds up: classify the purchase type and its characteristics first, identify the operational system that genuinely originates and evidences that transaction, and only then select the buy/pay flow the classification and the system can jointly support. Get the order wrong, pick the flow before understanding the system, and you are designing a control your own infrastructure cannot execute.



The contract is the missing input, not just the missing control

The stretch from signed contract to matched invoice has a specific failure point at its very origin. Once a contract is signed, its terms, rates, SLAs, agreed milestones, approved quantities, become the reference point every downstream control needs to check against. Most organizations that have invested in a contract repository, or even a Source-to-Pay suite with contract management built in, assume this problem is already solved. It usually is not, not fully.

The contract data is genuinely accessible when a purchase runs inside the suite's standard catalogue: pricing and terms get loaded as master data, and the three-way match or ERS flow can check against them automatically. The difficulty shows up for everything that does not originate through that catalogue, the maintenance contract billed against milestones, the subscription renewing outside the P2P tool, the service agreement negotiated separately from the procurement platform.

For these cases, the contract sits in a repository that is technically "there," but nothing in the transaction flow reaches into it automatically. Reconciliation and mismatch investigation fall back to a person opening the contract, or the SharePoint record, and checking it by hand.

This is why purchasing channel design cannot stop at "which system originates this purchase and which flow does it support." It must ask a second question: can this flow reach back to the contract as its source of truth at the moment a mismatch needs investigating, or does that reach-back still depend on someone opening a different system and reading it manually. Where the answer is the latter, the leak is not in the approval flow at all. It is in the space between the repository and the transaction.

Why this is a finance problem, not a procurement diagram

For a CFO, the payoff of getting this alignment right shows up in three places:

1

Working capital improves

Payment terms and timing are governed by flows that match how the business transacts, not by a generic policy that forces early manual intervention.

2

Leakage closes

Exceptions become rare rather than routine, the invoice that should not need a human does not get one.

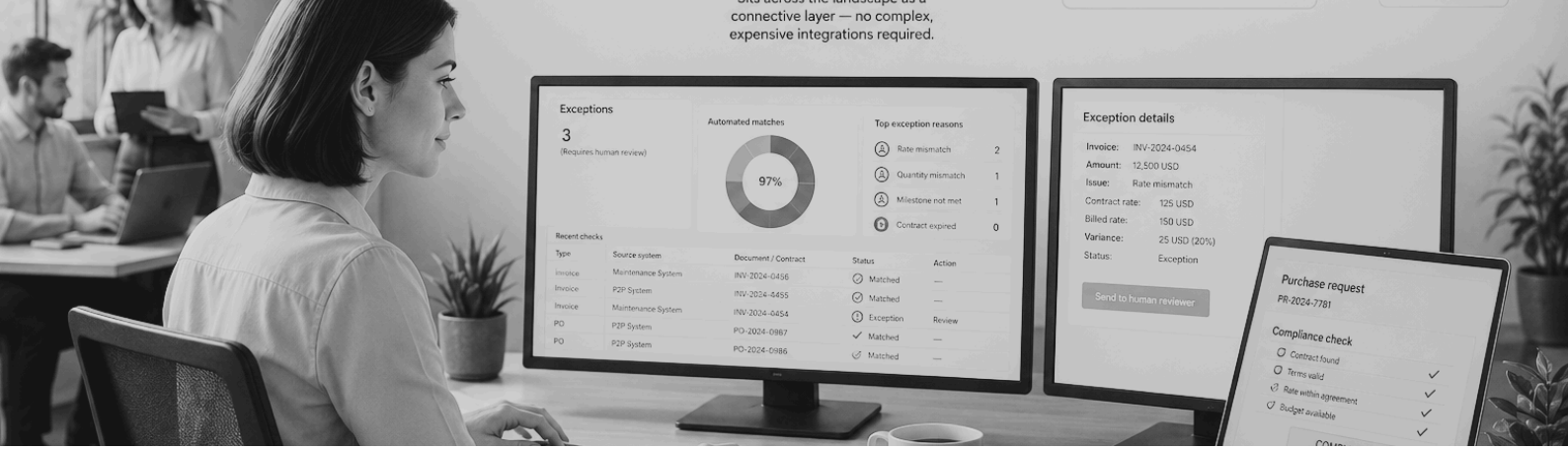
3

Cost-to-serve drops

In the accounts payable and procurement operations teams, because people are only pulled into the transactions that genuinely need judgment, not the ones a badly matched control forced into a queue.

None of this is a one-time fix. New vendors, new categories, new systems (an ERP migration, a new maintenance platform, an M&A integration) constantly reshape which purchase types exist and which systems capture them.

That is why this classification work needs to sit inside a real governance cycle, optimize, design, model, execute, monitor, with clear process ownership, rather than being a slide from a project that finished eighteen months ago. Total spend, broken down by channel, should be something the organization can look at and answer, for every slice: which system produced this, and does the control around it match what that system can prove.



Where AI agents change the equation

The gap described above, contract terms that exist but aren't reachable inside the flow, systems that were never built to talk to each other, integrations that are technically possible but expensive enough that nobody ever built them, has historically been closed by throwing more people or more integration budget at it. AI agent technology is starting to offer a third option.

An agent that can read a contract, understand what it authorizes, and check a specific transaction against it does not need the contract repository, the maintenance system, and the P2P platform to have been integrated in advance. It can sit across the landscape and do, on demand, what a person currently does by hand: pull the relevant contract, check the milestone, rate, or quantity, flag a genuine mismatch, and let everything that matches through untouched.

In addition to the 'downstream' check of invoices, an agent which checks for compliant commitment and contract terms 'upstream' before a PO is issued can support the process of securing that orders are not non-compliant with the agreed contract terms.

Framed this way, the agent is not a chatbot bolted onto procurement, it becomes part of the system landscape itself, a connective layer wherever a proper integration is technically feasible but has never been worth the cost to build.

This does not remove the need for the classification work described above, an agent still needs a purchasing channel map that tells which system to trust for which purchase type, and where a genuine exception should still go to a person, not a model. What changes are the economics of closing gaps between legacy systems that, until now, organizations have kept deferring for the next ERP cycle. For clients carrying a contract repository, a maintenance platform, and a P2P suite that were never designed to talk to each other, this is the first realistic opportunity to close that gap without waiting for a multi-year integration program.

Where the value actually gets won

This is precisely the stretch our P2P execution work is built around, the distance from signed contract to matched invoice, where value is won or lost in the details of buying channels, rate structures, and exception handling.

Getting purchasing channel classification right, anchored to the systems that actually originate each transaction, wired back to the contract as the source of truth, and increasingly supported by AI agents where integration has never been worth building, is the foundation that makes the rest of that engine run:

Ready to review how your purchasing channels are set up?

Contact Carve to learn how our P2P execution work can close the leak in your organization.

Jesper Sten Madsen
(+45) 6122 6102
jsm@carve.dk