AGENTIC PAYMENTS DEEP DIVE

Why Agentic Payments Might Finally Make IoT Payments Work

How regulated stablecoins, open payment protocols, and the NEXUS execution layer create the conditions for autonomous, governable machine-to-machine payments.

Quantoz | February 2026 | quantoznexus.com

Summary

Software agents, autonomous programs that act on behalf of users or devices, are rapidly evolving from “agents can decide” to “agents can transact.” The missing part has been payments: services need a standard way to request payment, and agents need a standard way to prove they are authorised to pay.

Two emerging open protocols, x402 and AP2, are now solving this by standardising how payments are requested, authorised, and proven, making agentic payments practical at scale.

Combined with regulated euro stablecoins like EURD and an execution layer like NEXUS, agentic payments can become controllable and auditable rather than ad hoc.

This article focuses on the technical and operational building blocks; it does not address legal or regulatory requirements for agentic payments, which can vary by jurisdiction and are still evolving.

1. From connected cars to paying pallets

For years, presentations about the “Internet of Things” (IoT) have promised that connected devices would not only send data but also pay for what they use. A connected car that automatically pays for charging, parking or tolls is now a standard example in the automotive world. Porsche, together with Quantoz, for instance, has publicly explored blockchain payments in the vehicle as a basis for new in-car services and commerce models.

In logistics, the story is similar. BASF has worked on “smart pallets” that continuously report their location, movement and loading status to improve supply-chain transparency. BASF also worked with Quantoz on adding a blockchain wallet to the smart pallets; if the pallet can already tell where it is and what it is doing, why shouldn’t it also be able to pay for warehousing, handling, charging or road use by itself?

This article first looks back at why early IoT payment initiatives stalled, then explains how regulated stablecoins such as EURD change the “money” side of the equation. It then introduces x402 and AP2 as emerging standards for how services request payment and how agents are authorised and held accountable. Next, it explains where NEXUS fits as the execution and system-of-record layer, before walking through two end-to-end examples: smart pallets and a consumer booking deposit.

2. Why the first IoT payment wave stalled, and why now is different

The early IoT payment pilots ran into two hard constraints.

First, there was no serious “money on-chain” for businesses. Most tokens were either volatile, unregulated, or both. Treasurers, auditors and regulators were reluctant to let machines move such assets around. What was missing was something that looked and behaved like electronic money, but lived on public blockchains: fully backed, regulated and redeemable at par value in fiat.

Second, there was no standard protocol for payments between machines and services. Every project ended up inventing its own payment infrastructure. The result was expensive point-to-point integrations with no network effect.

Today, both gaps are being closed. On the money side, Quantoz Payments issues Electronic Money Tokens (EMTs) under the EU's Markets in Crypto-Assets (MiCA) framework. EMTs such as EURD are regulated digital euros, designed to maintain a stable value of one euro each. They are fully backed by safeguarded reserves held with Tier 1 financial institutions and are issued under Dutch central-bank supervision. Depending on an organisation's accounting policy and auditor assessment, these tokens may be treated as cash-equivalent.

On the standard side, new open protocols have emerged that are designed from the ground up for agents and machines: x402 repurposes the HTTP status code 402 "Payment Required" to standardise how services request a payment and how agents return proof of payment within web and API calls. AP2, the Agent Payments Protocol, is a payment-agnostic framework for agentic payments, built on cryptographic mandates and a shared rulebook.

Taken together, EMTs and these open standards create the conditions that were missing in the first IoT wave: trustworthy money on-chain and an agreed way for machines and agents to talk about payments.

3. NEXUS and Quantoz Payments' EMTs as the Digital Money Layer

NEXUS is the operational layer that turns EMTs into something businesses can use every day: it manages wallets and accounts, executes transactions, and maintains an authoritative record for reconciliation, reporting, and audit, so finance and compliance teams have a dependable system of record for EMT activity, rather than relying on the blockchain as the back office.

NEXUS is a cloud platform with a web portal and APIs, so it can integrate into existing banking, treasury, and compliance workflows without requiring an organisation to run blockchain nodes or build low-level wallet infrastructure. For organisations building IoT or agentic payment flows, this matters because it turns stablecoin payments into a payment system you can run in production. Payments can be executed under defined controls, and the resulting transaction data is available in consistent formats for downstream systems.

Quantoz Payments uses NEXUS to issue and operate its EMTs. This includes not only moving EMTs on-chain, but also supporting the full lifecycle that regulated digital money requires, such as issuance and redemption flows and the operational controls that sit around them. The result is that agentic or device-led payments can be built on regulated on-chain money while still fitting into familiar enterprise governance and reporting expectations.

4. The x402 / AP2 standard: a common language for agentic payments

Agentic payments require more than the ability to move money. The harder challenge is coordination: how a service asks for payment in a consistent way, how an agent proves a payment happened, and how everyone involved can be confident that the agent was allowed to make that payment. The x402 and AP2 protocols address these pieces in a complementary way: x402 standardises the payment request and proof protocol at the HTTP layer, while AP2 standardises authorisation and accountability through mandates.

x402 makes payments feel native to the web and to APIs by activating HTTP status code 402, “Payment Required,” as a structured way for services to express a price. In an x402 flow, an agent calls a service provider’s API as it normally would. If payment is required, the service responds with an HTTP 402 and a machine-readable payment request that specifies what needs to be paid, in which asset, to which destination, and within what constraints such as time windows. The agent then completes the payment and repeats the same API request, this time including proof that the payment was made. The service verifies that proof and, if everything checks out, returns the normal HTTP 200 response.

AP2, the Agent Payments Protocol, adds the governance layer around this interaction. It provides a payment-agnostic rulebook for how agents, merchants and payment providers express permissions, exchange evidence, and produce an auditable trail. AP2’s core concept is the mandate: a tamper-proof, digital permission slip that defines what an agent is allowed to do on behalf of a user or organisation, such as spending limits, permitted counterparties, and allowed payment methods. In practice, AP2 makes the question “may this agent pay?” explicit and verifiable, rather than leaving it implicit inside application code. Because AP2 is designed to be payment-agnostic, the same mandate logic can govern different payment options, including stablecoins and x402-style HTTP payments.

It is important to be clear about what these protocols do not solve on their own. x402 can standardise payment requests and proof of payment, but it cannot guarantee that a payment request points to the correct recipient. An agent can receive a perfectly well-formed 402 response with a valid payment request that nevertheless directs funds to the wrong address, whether by misconfiguration or malicious intent. AP2 improves safety by ensuring an agent only acts under an explicit mandate, but it also does not magically solve identity across domains. A mandate can say “approved merchants,” yet the system still needs a way to bind “this API endpoint” or “this merchant identity” to “these legitimate payment destinations.” That is why practical deployments need an additional trust mechanism, such as an approved-provider registry or verified payment endpoints, and why the execution layer should enforce counterparty constraints so that untrusted payment details cannot bypass policy.

Agentic payments also introduce a new compliance dimension that sits alongside traditional “know your customer” checks: Know Your Agent (KYA). In a governed setup, organisations do not only verify the human or the business behind an account; they also register and classify the software agents and IoT devices that are allowed to act on their behalf. KYA ties an agent to its mandates and expected behaviour, so policies, limits and monitoring can be applied at the agent level. It also makes incident response practical: if an agent is compromised or starts behaving unexpectedly, mandates can be revoked, the agent can be quarantined, and exceptions can be escalated without shutting down the entire system.

All of this reflects the right security assumption for agentic payments: an agent can still do the wrong thing even when mandates are valid and protocol proofs verify. The goal of combining x402 and AP2 is therefore not to “trust the agent,” but to make the flow standardised and governable. Figure 1 shows how these pieces connect end-to-end; the next section explains where NEXUS fits in that sequence as the execution layer and the source of transaction truth.

5. NEXUS inside x402 and AP2: execution, controls, and source of truth

While x402 and AP2 provide the language and governance for agentic payments, the actual movement of funds and the enforcement of rules require a robust execution layer. This is where NEXUS comes in, acting as the operational backbone that turns standardised payment requests into secure, auditable transactions. It can refuse transfers to destinations that are not approved, apply per-transaction and per-day limits, and requires additional checks above certain thresholds.

After executing the transfer, NEXUS returns clean transaction facts, such as whether the payment succeeded, the amount and asset (for example EURD), the recipient, and an on-chain reference such as a transaction hash. Those facts can feed back into the AP2 layer as ground truth, so it is clear which mandate authorised a payment and which agent executed it.

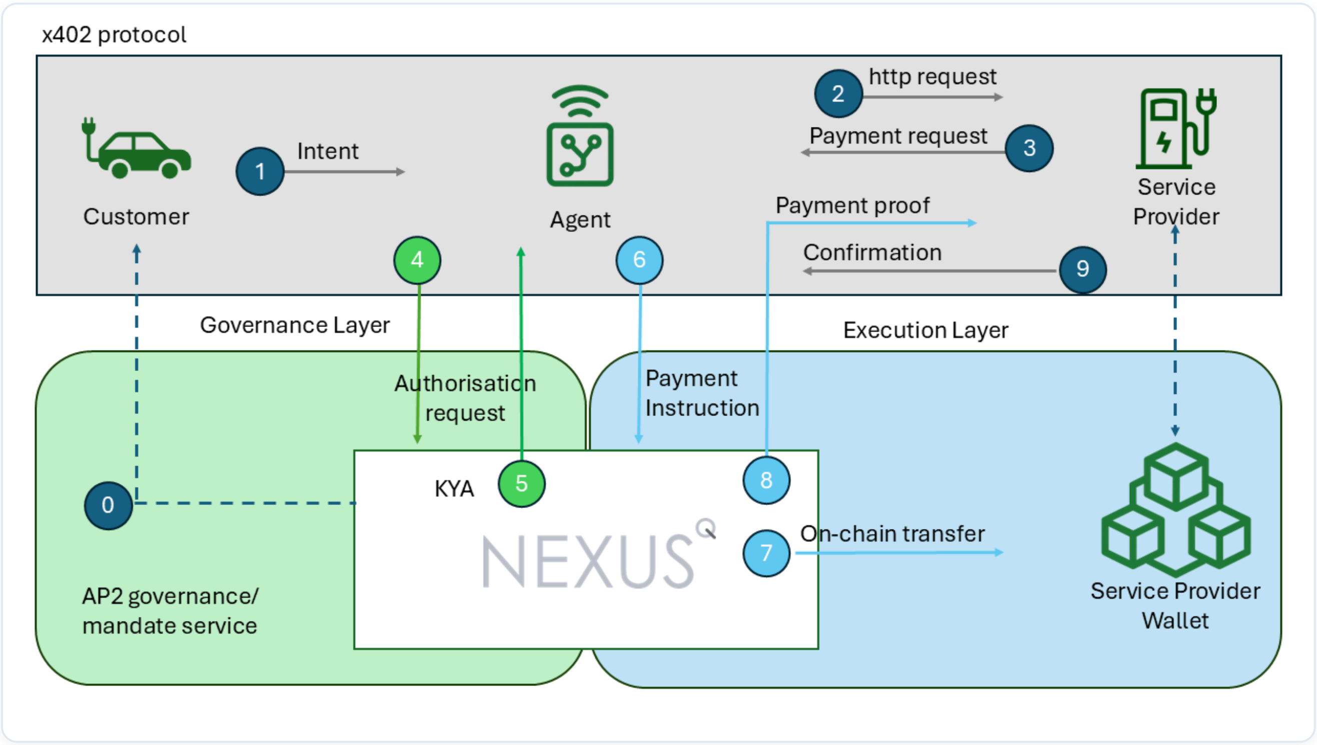

Figure 1. End-to-end x402 payment flow with AP2 governance and NEXUS execution, showing layered responsibilities from intent to on-chain transfer.

6. Autonomous smart pallets: from tracking to paying

With NEXUS and EMTs, a smart pallet can move from being merely trackable to being able to pay for services by itself, within clear limits. Imagine that each pallet has a NEXUS-linked wallet with a small operating balance, for example €25 in EURD, intended for incremental costs that are too small or too frequent to handle efficiently through invoicing. The logistics operator funds that wallet and defines a simple top-up rule, such as replenishing the balance back to €25 whenever it drops below €10, subject to whatever controls and approvals the organisation requires. Importantly, the pallet is not free to spend: AP2 mandates bound what it can pay for and with whom.

When the pallet arrives at a warehouse and requests storage, the warehouse service can respond with an HTTP 402 payment request that specifies the storage price for a defined period, the asset (for example EURD) and network, the destination address, and an expiry time. The pallet's agent does not treat that response as truth; it treats it as a request. It asks the AP2 governance and mandate service whether paying that amount for that service is permitted, and it also relies on the trust controls described earlier in this article.

If the mandate and counterparty checks succeed, the agent asks NEXUS to execute the payment. NEXUS performs the transfer, records the transaction, and returns a verifiable payment reference. The pallet then retries the original warehouse API call and includes proof of payment. The warehouse verifies the payment, either directly on-chain or via a trusted confirmation model, and confirms the service. For the logistics operator, the effect is practical and operational: a process that would otherwise involve invoices, approvals, exceptions, and reconciliation can become a controlled, near real-time pay-as-you-go flow.

7. Agentic consumer payments: a restaurant booking paid in EURD

The same pattern also applies when the "device" is not a pallet but an AI assistant acting for a consumer. Imagine a consumer saying, "Book a table for four at a good Italian restaurant at 19:30 and pay any reservation deposit in EURD."

In this scenario, the assistant is operating under AP2 mandates that might allow it to pay deposits up to a certain amount per booking and only to approved booking platforms or merchants. The booking platform will respond with an HTTP 402 "Payment Required". The assistant first checks that this payment fits the mandate and the applicable policy.

If it is allowed, the assistant asks NEXUS to execute the payment from the user's account to the platform or restaurant wallet. NEXUS executes and records the transfer and returns a payment reference that can serve as proof. The assistant then completes the loop by presenting that proof to the booking platform, which verifies the payment and confirms the reservation.

If something goes wrong later, there is a coherent record across intent, authorisation, execution, and outcome: the user's request, the mandate under which the agent acted, the payment instruction, the on-chain transfer, and the confirmation. From the user's perspective it feels like a single interaction with an assistant, but underneath it follows the governable, standardised sequence.

8. Developer and integration view: using NEXUS without becoming a blockchain shop

While these scenarios, from industrial pallets to consumer assistants, demonstrate the potential of agentic commerce, they also highlight a significant implementation challenge. Programming an agent to "pay for a service" is the easy part; operating the underlying infrastructure: safeguarding keys, tracking confirmations, and handling retries and failures, is the heavy lift for teams that are not blockchain specialists. NEXUS addresses this by separating the agent's business logic from the operational complexity of executing and monitoring on-chain payments.

In a combined x402 and AP2 setup, integration becomes a matter of standard API calls. A service provider can add an x402 “Payment Required” step to an existing HTTP API without redesigning its back office, while an agent developer can translate a payment request into a governed instruction by checking the AP2 mandate and then letting NEXUS handle execution. NEXUS returns a verifiable payment reference that completes the x402 handshake and produces consistent, auditable transaction data that fits into familiar ERP, reconciliation, and reporting workflows. Together, AP2 governance and NEXUS-managed execution let organisations deploy agentic payments without becoming blockchain operations teams.

9. From IoT payments to agentic commerce

What began as “machines paying for machine services” is becoming a broader pattern in agentic commerce, where software agents transact for physical services, digital services, and APIs in ways that look increasingly like normal web interactions. When services can express pricing directly in an API response and agents can complete payment and return proof, payments stop being a separate, bespoke integration and become part of the same request-response loop as the service itself.

This shift matters because the agentic internet is likely to create far more “small, frequent, context-specific” transactions than traditional commerce. Deposits, pay-per-use access, real-time microservices, machine-to-machine services, and automated renewals all benefit from the same capabilities: predictable payment requests, explicit permissions, and a reliable execution and record layer. With those building blocks in place, innovation shifts from reinventing payment rails to designing great products, sound policies, and strong partner relationships.

10. Why IoT payments might finally work this time

IoT payments can work at scale when autonomy is bounded and trust is explicit. The novelty is not that a device can send value, but that an ecosystem can agree on how payment is requested, how permission is granted, how payment is proven, and how the resulting financial facts are recorded. That combination is what turns a clever pilot into something an organisation can operate, control, and explain.

The practical implication for companies exploring connected devices, industrial APIs, or AI assistants is that the hard work has shifted. The question is no longer “how do we invent a payment stack that devices can use?” but “what is the safest, simplest first use case, and what policies do we want agents to follow?” In real deployments, the differentiator will be governance: the mandates you define, the counterparties you approve, the limits you enforce, and how you handle exceptions such as reversals, disputes, and compromised agents. With the right governance, autonomy becomes a feature rather than a risk.

Ready to explore agentic payments?

If you want to validate whether agentic payments make sense for your company, the most effective next step is a focused technical workshop around one concrete flow, such as pay-per-use access, deposits, or automated logistics services. Quantoz can help you map the end-to-end interaction, define mandate and counterparty requirements, and identify what would be needed to integrate an x402-style payment handshake and AP2-style authorisation with NEXUS-backed execution and reporting in your environment.

quantoznexus.com/agentic-payments