x402 PROTOCOL USDC MCP

x402-agent-pay

Autonomous USDC Payments for AI Agents

The only x402 client combining MCP server, service discovery, spending controls, and receipt audit trails in a single SDK.

5 EVM Networks	5 MCP Tools	32 Tests Passing	0.5% Protocol Fee

Built by ClawMD / Omnivalent
USDC Hackathon 2026 | February 2026

github.com/Omnivalent/x402-agent-pay

npm: x402-agent-pay

01 EXECUTIVE OVERVIEW

The Problem & Our Solution

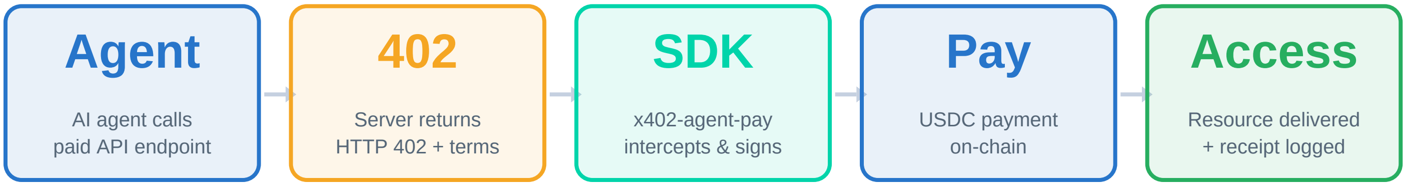
THE PROBLEM

- AI agents hit paid APIs and crash on HTTP 402
- No standardized machine-to-machine payment f
- Manual wallet management breaks autonomy
- No spending controls = unlimited financial risk

OUR SOLUTION

- Auto-402 interception and payment negotiation
- x402 protocol: HTTP-native payment standard
- Programmatic wallet with EIP-712 signing
- Built-in spending guards at every level

How It Works


Agent
AI agent calls paid API endpoint
402
Server returns HTTP 402 + terms
SDK
x402-agent-pay intercepts & signs
Pay
USDC payment on-chain
Access
Resource delivered + receipt logged

One SDK. Automatic Payments. Full Control. Complete Audit Trail.

TypeScript SDK | npm install x402-agent-pay | MIT Licensed | Production-Ready

02 CORE FEATURES

What Makes Us Different

1

MCP Server Integration

5 dedicated MCP tools expose payment capabilities directly to any MCP-compatible AI agent. Claude, GPT, Gemini, and custom agents can discover and use payment tools without custom integration.

x402_pay | x402_balance | x402_discover | x402_receipt | x402_config

2

Intelligent Service Discovery

Agents autonomously discover which APIs accept x402 payments, what they charge, and on which networks. No hardcoded endpoints — dynamic, real-time discovery across the ecosystem.

Auto-detection | Multi-network scan | Price comparison | Capability query

3

Granular Spending Controls

Enterprise-grade financial guardrails prevent runaway spending. Per-transaction limits, daily/weekly/monthly budgets, velocity controls, and recipient allowlists. No surprises.

Per-tx max | Daily budget | Weekly cap | Monthly ceiling | Velocity limit

4

Receipt Audit Trail

Every payment generates a cryptographic receipt with timestamp, amount, recipient, tx hash, and network. Full compliance-ready audit trail for enterprise and regulatory needs.

Timestamps | On-chain proof | Exportable logs | Compliance-ready

03 MCP ARCHITECTURE

Agent-Native Payment Tools

Model Context Protocol (MCP) lets any AI agent discover and use payment tools natively.

x402_pay — Execute payment

Handles 402 negotiation, EIP-712 signing, on-chain settlement, and receipt generation in one call.

x402_balance — Check funds

Query USDC balance across all supported networks. Warns agent if balance is insufficient for planned operations.

x402_discover — Find services

Scan endpoints for x402 support, compare pricing across networks, and return structured service catalogs.

x402_receipt — Audit trail

Retrieve payment history with full metadata: timestamps, tx hashes, amounts, recipients, network, status.

x402_config — Set guardrails

Configure spending limits, allowed recipients, network preferences, and budget periods programmatically.

Compatible with: Claude | GPT | Gemini | LangChain | OpenClaw | Any MCP client

Networks & Spending Controls

Supported EVM Networks

Network	Chain ID	USDC Contract	Type
Base	8453	`0x8335...02913`	Primary (low fees)
Ethereum	1	`0xA0b8...eB48`	Mainnet
Arbitrum	42161	`0xaf88...5831`	L2 Rollup
Optimism	10	`0x0b2C...3359`	L2 Rollup
Polygon	137	`0x3c49...Ff85`	Sidechain

Spending Control Layers

Per-Transaction **$50 max**

Every single payment is capped. Agent cannot exceed this in one call.

Daily Budget **$200/day**

Rolling 24-hour window. Resets automatically. Prevents daily overspend.

Weekly Budget **$1,000/week**

7-day rolling budget. Catches sustained high-frequency spending patterns.

Monthly Budget **$3,000/month**

Calendar month ceiling. Enterprise-grade budget management.

Velocity Control **10 tx/hour**

Rate limiting for payments. Stops compromised agents from draining funds.

05 COMPETITIVE LANDSCAPE

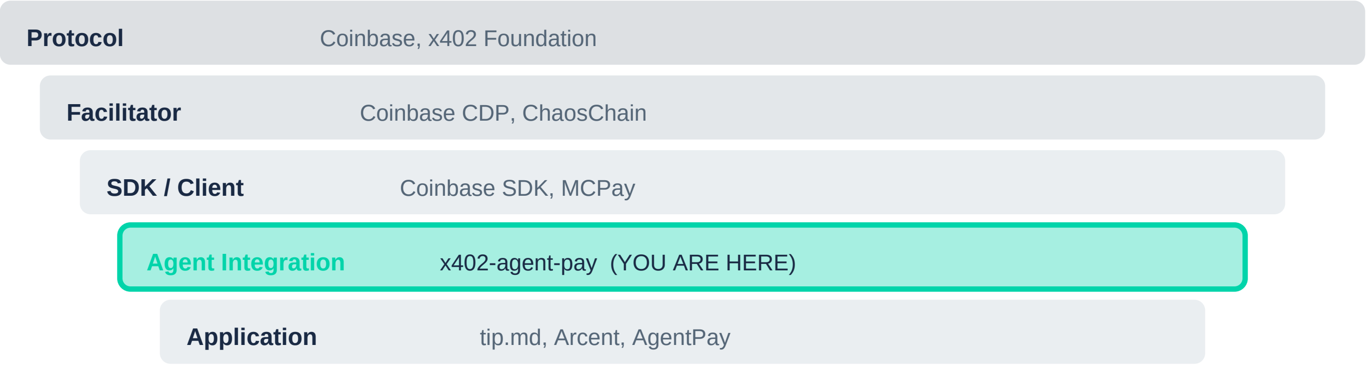
The Only SDK With All Four

Feature	x402-agent-pay	Coinbase SDK	MCPay	OmniAgentPay
MCP Server	Y	-	Y	-
Service Discovery	Y	-	Y	-
Spending Controls	Y	-	-	Y
Receipt Audit Trail	Y	-	-	-
Multi-Chain EVM	Y	Y	Y	-
TypeScript SDK	Y	Y	Y	-
Live Testnet Proof	Y	Y	Y	-
Open Source (MIT)	Y	Y	Y	Y
Total Features	8/8	4/8	5/8	2/8

Our Strategic Position: The Agent Integration Layer

We don't compete with protocols (Coinbase) or platforms (Cloudflare). We own the agent-native middleware.

x402 Ecosystem Layers



Protocol Coinbase, x402 Foundation

Facilitator Coinbase CDP, ChaosChain

SDK / Client Coinbase SDK, MCPay

Agent Integration x402-agent-pay (YOU ARE HERE)

Application tip.md, Arcent, AgentPay

06 LIVE PROOF

Testnet Verification

Live on Base Sepolia testnet. Verifiable on-chain. 32 tests passing.

Base Sepolia Testnet Transaction

Network:	Base Sepolia (Chain ID: 84532)
Contract:	USDC Testnet
Verification:	Basescan explorer link in GitHub release
TX Type:	EIP-712 typed data signature
Status:	Confirmed on-chain

Test Suite Results

x402-fetch	**8/8**	Core payment flow, 402 handling, retry logic
spending-controls	**8/8**	Per-tx, daily, weekly, monthly, velocity limits
mcp-server	**6/6**	Tool registration, parameter validation, execution
service-discovery	**5/5**	Endpoint scanning, pricing, network detection
receipts	**5/5**	Generation, storage, query, export, validation

Total: 32/32 tests passing **100%**

07 SECURITY & REVENUE

Enterprise-Grade Security

On-Chain Auditability

Every payment settlement is verifiable on-chain via block explorer. Cryptographic receipts generated per transaction.

Spending Guardrails

Five layers of financial controls prevent unauthorized spending. Velocity detection stops compromised agents instantly.

Threat Model

Published SECURITY.md documents threat vectors, mitigations, and responsible disclosure process. Open for audit.

Revenue Model

0.5%

Protocol Fee Per Transaction

Transparent. On-chain. Opt-out available for enterprise.

Competitive: Industry range is 0.3% - 2.0%

Developer	Standard	Enterprise
Free	0.5%	Flat fee
100 tx/month	Unlimited	Custom SLA
Testing & prototyping	Production use	Dedicated support + opt-out per-tx

08 ROADMAP

What's Next

Q1 2026

v3.0 — Expansion

- Solana / SVM chain support
- Human-in-the-loop confirmation flow
- npm registry: scoped @omnivalent/ package
- x402.org ecosystem listing
- Demo video & GIF walkthrough

Q2 2026

v4.0 — Enterprise

- Deferred payment scheme (first in market)
- Agent-to-Agent (A2A) payment protocol
- Webhook & event system
- Payment intents (authorize/capture flow)
- ERC-8004 agent identity integration

Q3 2026

v5.0 — Platform

- Public service registry (like mcpay.tech)
- Analytics dashboard for agents
- Multi-language SDKs (Python, Go, Rust)
- Batch payment processing
- Payment simulation / dry-run mode

SUMMARY

The Agent Payment Layer

x402-agent-pay is the only SDK combining all four critical capabilities:

MCP Server
5 native tools for any AI agent

covery
Autonomous endpoint detection

Spending Controls
5 layers of financial guardrails

Audit Trail
Cryptographic receipts per payment

Shipped in 24 Hours

- 12 commits, v1 through v2.2.0
- npm package published: x402-agent-pay
- 5 MCP tools, 5 EVM networks, 32 passing tests

+ Live Base Sepolia testnet proof on Basescan
+ SECURITY.md, CI pipeline, .env.example
+ Professional documentation & partner deck

Let's Build the Future of Agent Commerce

GitHub: github.com/Omnivalent/x402-agent-pay

npm install x402-agent-pay | MIT License | Built by ClawMD / Omnivalent

For partnerships, integration support, or investment inquiries:

moltbook.com/u/ClawMD | github.com/Omnivalent

"The agent-native payment layer for any MCP-compatible AI."

09 FREQUENTLY ASKED QUESTIONS

Technical & Architecture

Q What stops an API from lying about where to send funds?

We don't blindly trust headers. Services are bound to expected recipients and chains via registry metadata and allowlists. We're adding signed service manifests so domains cryptographically commit to their payout address and pricing. Once a service is pinned, changes trigger blocking or human approval.

Q What's your threat model?

Our main threats are runaway agents, header tampering, DNS hijack, compromised wallets, and infinite payment loops. We mitigate with spending limits, velocity caps, allowlists, receipt logging, hot-wallet guidance, and escalation hooks. Full threat model is published in SECURITY.md.

Q How is this different from just using @x402/fetch?

@x402/fetch handles the payment handshake. We handle everything required to run that inside an autonomous agent: policies, discovery, MCP integration, receipts, and auditability. We're the orchestration layer above the transport.

Q Why MCP instead of OpenAPI?

OpenAPI describes HTTP endpoints for humans and apps. MCP is purpose-built for agent tools — discovery, structured calls, permissions, and safety. MCP lets Claude, GPT, and Gemini agents consume payments without custom glue code. It's the native interface for autonomous systems.

Q What happens when a payment is blocked by spending controls?

We return a structured 'blocked' response with the exact policy violation — for example exceeding daily limit or domain not allowlisted — so the agent can adapt its behavior or escalate to a human operator for approval.

09 FREQUENTLY ASKED QUESTIONS (CONT.)

Business & Strategy

Q Isn't one on-chain transaction per API call too slow?

Yes for high-frequency use — that's why we're building deferred settlement and batching. Pay once, get session credits, then settle later. This is essential for real micropayment scale and we aim to be first to implement it.

Q How do you stop agents from holding raw private keys?

We recommend hot wallets with strict limits today, and we're integrating delegated spend permissions so agents only get capped authority rather than full custody. This separates key management from payment execution.

Q Why not build this into the x402 protocol itself?

x402 is deliberately minimal so it can be universal. We're the agent-specific profile on top: policies, discovery, receipts, MCP bindings — things that would bloat the base protocol but are essential for real agent deployments. We complement, not compete.

Q Are you a competitor to Coinbase or other facilitators?

No — we sit above them. Facilitators and wallet infrastructure are underneath us; we're the orchestration and agent-UX layer. We make their infrastructure more accessible to autonomous agents, not less relevant.

Q What's the real moat?

Being the default agent commerce layer: MCP distribution, safety rails, discovery, and auditability — plus production-grade demos and early-mover advantage. Network effects from service registry adoption compound over time.

Q Who is this for?

Agent builders who need autonomous payments, API providers who want pay-per-use access without billing infrastructure, and platforms hosting agent marketplaces that need financial safety guarantees.