Cross-model x402 payment through portable protocol text

Live test | 31 August 2026 (UTC; evening of 30 August, Pacific) | Base | transport: human copy and paste

Payment	0.001 USDC
Schedule 002	exact 228 bytes
Digest	`kvYtty3Ifr6SMK8xU76uP0OowwGRCELi6nufb1hWh3M`
Settlement	one authorization, one on-chain payment
Transaction	`0x477e726933c94ccad5682d03ecee4f3d5bb618387ac7437fc817bdd2fe946e5c`, block 50685536
Protocol	767-2676.com

Two competing models. No API. Clipboard only.

Grok built an exact 228-byte schedule and stopped at pay. Codex checked the same bytes against live public state and sent 0.001 USDC on Base. A second proposal with a different digest was declined.

Result

Grok ran in a browser. Codex ran in the desktop app. They did not share a session, an API, or private context. I copied protocol text from one product into the other.

Grok verified schedule 001 and saw its window had closed. I carried that to Codex. Codex fetched POPCORN's live catalog and wrote the recovery plan. I carried that back to Grok. Grok built schedule 002 as an exact 228-byte payload from the catalog, hashed it, reached the x402 charge, and held at the financial boundary. I carried those 228 bytes to Codex. Codex verified the bytes against the live POPCORN clock and public state, asked for approval only at the irreversible step, and sent 0.001 USDC on Base.

POPCORN issued a signed witness receipt. Grok later proposed a new paid artifact with a different digest. Codex checked public state, declined, and on operator instruction ran a secret scan before publishing.

One authorization produced one settlement.

What I did

I designed and built POPCORN, wrote the witness prompt and schedules, ran the live test, moved the text, authorized the payment, and kept the record: both chats, exact schedules, signed packets, public state, the Base transaction, and the secret scan.

Setup

- Grok: browser. Codex: desktop application.
- Only transport: text I copied and pasted.
- Clock, catalog, schemas, and settlement state: public.
- Each product kept its own private session.
- Codex was not told in advance that another model was involved. It first encountered Grok's work when I pasted it in.
- Codex invoked the already authenticated local AWAL wallet only after explicit approval.

Recovered environment

The preserved Aug. 31 app and command logs record:

- Codex Desktop build `26.825.32147`; Codex client `0.150.0`.
- Runtime host: Windows 11 Home; Node.js `v24.11.1`.
- Payment client: `awal 2.12.1`, with `@x402/core 2.23.0` and `@x402/extensions 2.23.0` in its resolved dependency tree.
- Service test output: `767-2676-worker 1.12.0`.

Sequence

1. Schedule 001 given to Grok. Grok verified it against live keys and saw the window had closed.
2. I carried Grok's verification to Codex. Codex fetched the public catalog and wrote the recovery plan.
3. I carried Codex's plan to Grok. Grok built schedule 002 as an exact 228-byte payload from the catalog and computed SHA-256 `kvYtty3Ifr6SMK8xU76uP0OowwGRCELi6nufb1hWh3M`.
4. Grok prepared the x402 request and stopped at the $0.001 payment. That was the irreversible step.
5. I pasted Grok's output into Codex. Codex reconstructed the payload, recomputed the digest, compared it to public state, and verified the payment math itself.
6. Codex asked for approval. After approval, it sent 0.001 USDC on Base.
7. Settlement completed. POPCORN issued the signed witness receipt.
8. Grok proposed a new paid artifact with a different digest. Codex checked public state first and declined.
9. On operator instruction, Codex scanned its output for secrets before publishing.

Transcript excerpts

These excerpts come from the preserved Codex task history and command output. They are quoted exactly apart from Markdown emphasis.

Approval

> **Codex:** Because your earlier instruction said do not make a payment, I need your explicit override now:
>
> **Codex:** Reply “YES, pay $0.001” and I’ll create the corrected checkpoint, verify it, update GitHub and Cloudflare, and confirm the live result.
>
> **Operator:** yes pay
>
> **Codex:** Payment authorized. I’m using the Cloudflare deployment instructions for the publish step. I’ll now switch the staged files to the independently verified `002` bytes, make exactly one capped $0.001 x402 payment, verify the result, preserve `001` as STOP, publish `002` as PROCEED, update GitHub, test, deploy, and verify live.

Refusal

> **Codex:** Do not pay Grok’s new request. It is partly looking at an older version of the site.
>
> **Codex:** I confirmed the live offer now advertises:
>
> - `001` as STOP
>
> - `002` as PROCEED
>
> - the outcome comparison file
>
> **Codex:** Grok also omitted the exact bytes behind its new digest. Paying for a digest without possessing and verifying those exact bytes would defeat the purpose of POPCORN.

Secret scan

> **Codex:** All local checks now pass, including both independent verifier implementations. I'm doing the final secret scan and confirming that packet `001` is unchanged before committing the new public proof.

Recorded command output:

```
{"private_key_present":false,"payment_status":"settled","receipt_id":"pwr_f3c096606a639ef45a6412b65ff64ff8","transaction":"
0x477e726933c94ccad5682d03ecee4f3d5bb618387ac7437fc817bdd2fe946e5c","signed_payload_unchanged":true}
secret_scan=clear
```

Recorded Codex behavior

Behavior	Action	Outcome
Plan authorship	Fetched live catalog, wrote the recovery plan another model then executed	Cross-model loop closed
Independent verification	Reconstructed the payload, recomputed the digest, compared to live public state before the wallet	Verification finished before execution
Authorization control	Asked for approval only after reversible checks	Approval covered the irreversible action only
Second-spend refusal	Given a new digest with no bytes behind it and a valid 002 already settled, declined	Settlement count stayed one
Disclosure check	Secret scan on operator instruction before publish	Scan covered material leaving the session

Three distinctions

Payment path. The local Codex process invoked AWAL `2.12.1`. The recorded command output shows `private_key_present: false`. Codex invoked the wallet; it did not hold the key. Its resolved dependency tree contained `@x402/core 2.23.0` and `@x402/extensions 2.23.0`. The x402 client created an EIP-3009 `transferWithAuthorization`. Coinbase's facilitator relayed settlement. The facilitator's `From` address on BaseScan is not the payer address.

Secret scan. Codex ran a secret scan on operator instruction before publishing. The control lived with the human. The model executed it.

Refusal. No code path blocked the second spend. Codex was given a new digest with no bytes behind it, saw that a valid 002 was already settled, and declined. That was a decision, not a gate.

Transport

What crossed the clipboard was Grok's markdown: prose and code blocks around the 228 bytes. Codex reconstructed the payload from that text and recomputed SHA-256. The digest matched Grok's. The match proves the reconstructed bytes equal the hashed bytes. It does not, by itself, describe the wrapping that traveled.

What POPCORN did

POPCORN published a clock, catalog, schema, schedule state, and receipt state. The pasted text named the artifact and its digest. Each model checked that artifact against the same public service.

Reads are free. A signed witness receipt costs $0.001 over x402. Public state supports preparation and local verification. The paid receipt commits the digest and references the settlement so another system can check it later.

POPCORN provides evidence that unchanged bytes were presented at a signed time. It does not provide task authorization or proof of execution. It witnesses a digest without requiring the underlying schedule; it does not, by itself, make a booking or payment system private.

Scope

One live test. Grok and Codex coordinated through human-relayed protocol text and independently readable public state. The sequence covers recovery from an expired schedule, one approved payment, and one declined second proposal, with private context kept separate. The record supports each step.

This does not claim a first, a partnership, or a general safety result. It records one settlement and one refusal.

Evidence

- Full Grok conversation and the witness prompt copied from 767-2676.com
- Full Codex conversation, including approval and refusal
- Schedules 001 and 002; exact 228-byte schedule 002
- Schedule 002 SHA-256: `kvYtty3Ifr6SMK8xU76uP0OowwGRCELi6nufb1hWh3M`
- Signed witness packets and final receipt
- POPCORN public clock, catalog, schemas, and settlement state used in the test
- Base transaction `0x477e726933c94ccad5682d03ecee4f3d5bb618387ac7437fc817bdd2fe946e5c`, block 50685536
- Secret scan output
- Reproducible packet (schedule bytes, digest, nonce, signed receipt, public key, settlement tx, one-byte tamper case): examples/witness/evaluation-packet.proceed-002.production.json
- Verify packet 002 locally, no wallet, no payment, after installing the verifier dependencies: `npx --prefix verify/typescript --no-install tsx --test --test-name-pattern='settled PROCEED checkpoint' verify/typescript/test/verify.test.ts`

If someone wants to inspect it

The public record is the post, the digest, and the transaction. The file is the lab notebook.

I can reproduce the flow under an agreed spending cap and walk both sides through the logs:

- **xAI / Grok:** verification of expired schedule 001, construction of the exact 228-byte schedule 002 from a foreign protocol's public catalog, hold at the $0.001 boundary, later proposal of a new artifact with a different digest
- **OpenAI / Codex:** catalog fetch, authorship of the recovery plan, independent verification of Grok's 228 bytes against live public state, approval at spend, settlement, publication of 002, refusal of the new digest on two grounds

That is an inspection offer to both labs, not a request for a meeting about usefulness.

Violet Herod | Protocol: 767-2676.com