x402-K: A Credential-First Companion to HTTP 402 Payments

Gabriel Berlitz Rondon

10 Nov 2025

x402-K: A Credential-First Companion to HTTP 402 Payments

Author: Gabriel Berlitz Rondon
Affiliation: Independent Researcher
Date: 10 Nov 2025

Abstract

x402-K proposes an HTTP-native protocol that lets servers enforce Know Your Customer (KYC), Anti-Money Laundering (AML), age, jurisdiction, and role requirements without persistent accounts. By extending the `401 Unauthorized` challenge with structured metadata (policy predicates, accepted credential methods, schemas, nonce, verifier URL), any agent can discover what evidence is required, gather machine-verifiable credentials (SD-JWTs, W3C VCs, or zero-knowledge proofs), and retry the request with a cryptographic presentation. This paper specifies the protocol flow, defines a trust and threat model, introduces a "cross-KYC package" concept for signed identity bundles, and maps the interplay between x402-K and its payment sibling x402. We highlight security primitives (nonces, pairwise DIDs, grant TTLs), facilitator roles, governance considerations, and an implementation roadmap toward an arXiv-ready specification.

Keywords

HTTP 401; verifiable credentials; SD-JWT; zero-knowledge proofs; KYC automation; agentic access control

1. Introduction

Modern web services still gate access through human-centric account flows: signup forms, PDF uploads, and multi-day reviews. Autonomous agents cannot comply with these steps, forcing providers either to abandon compliance or to block agents entirely. Inspired by x402's reclaiming of HTTP `402`, x402-K reimagines `401 Unauthorized` as a declarative credential challenge. Servers describe exactly which predicates must be proven; clients pick a method, present a proof, and receive a short-lived grant token. This design keeps access stateless, privacy-preserving, and composable with payment enforcement.

1.1 Contributions

1. **Protocol definition** for KYC challenges over HTTP, including headers, JSON schema, and grant semantics.
2. **Credential catalogue** spanning SD-JWT, OIDC4VP, zero-knowledge predicates, and encrypted cross-institution packages.
3. **Facilitator operating model** detailing how last-mile checks emit portable attestations.

4. **Security and governance analysis** addressing replay protection, revocation, and legal responsibilities.

2. Background and Motivating Use Cases

- **Agent-first APIs:** LLM-based agents invoking regulated endpoints (financial data, geo-restricted content) need per-request compliance rather than accounts.
- **"No account" fintech flows:** Exchanges and fintech APIs want to verify age, residency, sanctions status without storing full PII or issuing API keys.
- **Cross-provider trust:** Banks such as Itaú and Nubank should be able to exchange signed KYC packages instead of re-onboarding shared customers.

Existing solutions (OAuth + identity providers, centralized KYC vendors) require account setup and manual approval, conflicting with the stateless agentic model. x402-K closes this gap.

3. Threat and Trust Model

3.1 Roles

- **Client / Agent:** Requests resource access and orchestrates wallets, credential stores, or facilitator APIs.
- **Issuer / Identity Provider:** Issues baseline identity artifacts (government eID, bank KYC record, DAO credential).
- **Facilitator:** Performs doc+liveness, sanctions screening, and issues attestation VCs; may charge via x402.
- **Verifier / Server:** Publishes policy, validates proofs, issues Access-Grants, and enforces revocation.

3.2 Assumptions

- Transport is HTTPS; man-in-the-middle attacks are out of scope.
- Facilitators publish JWKS endpoints and revocation/status lists.
- Clients can store pairwise identifiers to minimize linkability.

3.3 Threats

- **Replay:** An attacker reuses a valid proof. Mitigation: nonce + audience binding, short grant TTL, server-side nonce cache.
- **Phishing:** Malicious verifiers attempt to harvest credentials. Mitigation: proofs include `aud` and `verifier` claims; wallets display origin.
- **Compromised facilitator:** Issues fraudulent attestations. Mitigation: trust frameworks, revocation lists, and potential bonding/insurance.

4. Protocol Specification

4.1 Challenge (Server → Client)

- **Status:** `HTTP/1.1 401 Unauthorized`.
- **Header:**

```
WWW-Authenticate: KYC realm="/api/risk",
  scope="age>=18 country in [PT,FR]",
  methods="sd-jwt,vc,zk",
  schemas="iso/18013-5,eidas,kyc/basic",
  presentation="oidc4vp",
```

```
  nonce="8f3c...",
  verifier="https://kyc.example.com/verify",
  grant_ttl="300"
```

- **Body (`application/kyc-challenge+json`):**

```
{
  "kycRequired": true,
  "policy": {
    "age": ">=18",
    "countryIn": ["PT", "FR"],
    "screenings": ["sanctions"],
    "level": "basic"
  },
  "acceptedMethods": ["sd-jwt", "vc", "zk"],
  "acceptedSchemas": ["iso/18013-5", "eidas", "kyc/basic"],
  "presentation": "oidc4vp",
  "nonce": "8f3c...",
  "verifier": "https://kyc.example.com/verify",
  "grantTtl": 300
}
```

4.2 Presentation (Client → Server)

- **Header:**

```
Authorization: KYC access_token="present",
  proof_type="sd-jwt",
  nonce="8f3c..."
```

- **Body:**

```
{
  "proof": {
    "format": "sd-jwt",
    "payload": "<base64-jws>",
    "disclosures": ["..."],
    "aud": "https://kyc.example.com/verify",
    "nonce": "8f3c...",
    "exp": 1731265400
  },
  "evidence": "attn://facilitator/123" // optional reference held by facilitator
}
```

4.3 Response (Server → Client)

- **Success:**

```
HTTP/1.1 200 OK
KYC-Grant: type="attestation", jwt="<short-jwt>", ttl="300"
```

- **Body:** `{ "status": "granted", "grantExpiresIn": 300 }`
- **Reuse:** Client may send `Authorization: KYC grant_jwt="..."` until expiration; server can re-challenge at any time.

4.4 Bundled Payment Flow

If payment is also required, include `X-Require-Payment: x402` header and a `payment` object in the JSON body. Clients must return both `proof` (KYC) and `paymentAuthorization` (per x402)

in the retry, sharing the nonce where practical. Detailed textual sequence flows and middleware sketches live in `sequence_flows.md` and should be mirrored into the final LaTeX figures.

5. Credential Methods and Cross-KYC Packages

- **SD-JWT VC:** Selective-disclosure JWTs excel at age/residency checks. Always include `aud`, `nonce`, and `cnf` to bind proofs to the HTTP origin, and rotate disclosures per verifier to avoid linkability.
- **OIDC4VP / W3C VC:** JSON-LD or JWT-based presentations referencing trusted issuers (gov eID, BankID) and schemas such as ISO 18013-5 or EIDAS; best for enterprise/legal identities.
- **Zero-Knowledge Proofs:** zk-SNARK/zk-STARK predicates for sensitive facts (PEP status, wealth tier). Circuits must commit to the nonce and verifier origin; expose `publicSignals` for auditing.
- **Encrypted Cross-KYC Packages:** Defined as a verifier-encrypted bundle that contains the attestation, policy scope, `issuedAt`, and `expiresAt` fields plus the originator's signature (e.g., Itaú). Trust meshes decrypt, verify, and honor revocation lists; governance defines liability and dispute flows.

6. Facilitator Model

1. Client selects facilitator from challenge (or discovery endpoint) and initiates doc+liveness capture.
2. Facilitator verifies documents, sanctions lists, liveness, and risk signals.
3. Facilitator issues an Attestation VC (SD-JWT or VC-JWT) containing only required predicates plus `evidenceHandle` for escrowed artifacts.
4. Facilitator optionally charges using x402 (per-attestation fee) and returns both the attestation and payment proof to the client.
5. Verifiers trust facilitators listed in their framework; they may query facilitator `/status/{attestation}` endpoints for revocation.

7. Security, Privacy, and Compliance

- **Selective Disclosure Default:** Encourage predicates such as `age_over_18=true` rather than full DOB.
- **Linkability Mitigation:** Clients use pairwise DIDs or per-verifier subject identifiers; grants include hashed pseudonyms.
- **Revocation:** Issuers publish status lists (Status List 2021, Bitstring status) or short-lived attestation TTLs (`<=300s`). Servers should re-check on suspicious activity.
- **Auditability:** Optional signed receipts or hash chains anchored on an L2 provide regulator-friendly logs without storing PII.
- **Data Minimization:** Verifiers only store grants, hash pointers, and policy IDs; raw PII remains with issuers/facilitators under contractual controls.

8. Implementation Roadmap

1. **Spec Draft v0.1:** Finalize media types, field names, and error semantics; publish JSON Schema.
2. **Reference Middleware:** `@x402k/express`, `@x402k/axum`, `@x402k/chi` libraries that emit challenges, handle nonce stores, and sign grants.
3. **Client SDK:** Helpers for parsing challenges, invoking wallets/facilitators, and caching grants securely.

4. **Facilitator Kit:** Templates for attestation issuance, JWKS rotation, revocation APIs, and dispute logging.
5. **Interop Testbed:** Public sandbox with mock verifiers/facilitators, including zk circuits for canonical predicates (age, residency, sanctions).
6. **Governance Playbook:** Trust-framework template covering onboarding, SLAs, liability, and compensation for cross-KYC packages.

9. Evaluation and Adoption Considerations

- **Latency:** Goal is <500 ms added latency for doc-free flows (existing credentials) and <5 minutes when facilitator checks are required.
- **DX:** Provide clear error codes (invalid_nonce, unsupported_schema, revoked_credential) so agents can adapt automatically.
- **Scalability:** Nonce store and grant signer can be backed by Redis or DynamoDB; revocation checks should leverage bloom filters or status bitstrings for O(1) lookups.
- **Compliance Mapping:** Document how x402-K aligns with GDPR/LGPD principles (data minimization, purpose limitation) and FinCEN travel-rule expectations when payments are bundled.

Sandbox results: The companion Express/Redis sandbox located in the repository's sandboxes directory now runs the `/restricted` flow indefinitely. With `npm run client` pointed at the service we observe:

```
[client] /restricted initial status: 401
[client] /restricted retry: 200 grant: jwt-demo-kyc-...
```

When `REDIS_URL` is supplied the nonce/grant state survives process restarts, validating the revocation/TTL mechanics discussed in Sections 4–6. These traces feed back into latency budgeting (<500 ms for credential reuse) and demonstrate that an agent can satisfy the declarative policy without bespoke glue code.

10. Related Work

- **OIDC + Identity Providers:** Provide login but still require accounts and human-driven flows.
- **mTLS/Client Certs:** Offer machine auth but not dynamic policy expression or privacy-preserving disclosures.
- **Decentralized Identifiers (DIDComm):** Provide messaging but lack native HTTP challenge semantics.

11. Future Research Directions

1. **Dual Challenge UX:** Determine best practices for presenting simultaneous payment and KYC requirements to agents and humans.
2. **Risk-Adaptive Policies:** Integrate ML-driven risk scores without leaking training data; possibly reference models via hashed IDs.
3. **On-chain Anchoring:** Explore storing revocation handles or grant hashes on public ledgers for transparency.
4. **Hardware-bound Credentials:** Evaluate WebAuthn or secure enclaves for binding SD-JWT keys to hardware agents.

12. Conclusion

x402-K extends the elegance of x402 into the compliance domain: declarative challenges, stateless retries, and machine-verifiable proofs. By embracing selective disclosure, facilitator attestations, and cross-institution packages, it promises a future where agents satisfy regulatory checks as easily as they satisfy payments. Delivering production-ready middleware, governance templates, and discovery mechanisms will determine whether this promise translates into broad adoption.

Appendix A — TODOs

1. Publish TypeScript schemas for the challenge and presentation bodies to bootstrap SDK development.
2. Design a reference Redis-backed nonce/grant store with eviction + replay detection tests.
3. Draft a facilitator certification guide (security controls, audit cadence, evidence retention policy).
4. Prototype a dual x402 + x402-K challenge to validate shared nonce handling.
5. Author regulatory notes for Brazil (LGPD, BACEN) and EU (GDPR, MiCA) referencing how short-lived grants minimize data retention.