CONFIDENTIAL

# UNITED KINGDOM INTELLECTUAL PROPERTY OFFICE

# PATENT APPLICATION

## Applicant

**The Bitcoin Corporation Ltd**

## Title of Invention

**HTTP Status Code Token Protocol Suite: A System and Method for Mapping HTTP Authentication, Payment, and Authorisation Status Codes to Blockchain-Native Identity, Commerce, and Securities Token Layers**

## Field of the Invention

The present invention relates to systems and methods for tokenising web resources using blockchain technology, wherein the semantics of HTTP status codes are mapped to distinct blockchain token protocol layers. More particularly, the invention concerns a tripartite protocol suite in which HTTP 401 (Unauthorized) maps to an on-chain identity token system, HTTP 402 (Payment Required) maps to a content commerce token system with algorithmic pricing and Proof of Indexing consensus, and HTTP 403 (Forbidden) maps to a programmable securities and access control token system, the three layers operating as an integrated stack wherein each layer may reference and depend upon the others.

## Background of the Invention

### Problem Statement

The World Wide Web uses HTTP status codes to communicate the state of a request between client and server. Three codes in particular — 401 (Unauthorized), 402 (Payment Required), and 403 (Forbidden) — describe escalating levels of access denial. Yet the web has no native mechanism to resolve these denials through cryptographic proof:

1. **HTTP 401 — Identity Without Portability** — When a server returns 401 Unauthorized, it requires the client to authenticate. Current solutions (OAuth, SAML, OpenID Connect) produce identity assertions that are: (a) scoped to a single service provider, (b) revocable by the identity provider at any time, (c) not composable across providers, and (d) not independently verifiable by third parties without contacting the issuing authority. There is no mechanism for a user to accumulate a portable, self-sovereign, multi-provider identity proof that any server can verify without relying on the original identity providers.

2. **HTTP 402 — Payment Without Settlement** — HTTP 402 Payment Required was reserved in HTTP/1.1 (RFC 2616, 1997) for future use with a digital payment mechanism. Nearly thirty years later, no standardised protocol has been implemented to give this status code operational meaning. The web was designed to support native payments but the infrastructure was never built. No standardised protocol exists for: (a) embedding machine-readable pricing terms in a 402 response, (b) enabling programmatic payment and retry without human authentication steps (CAPTCHA, 3D Secure, manual card entry), (c) minting transferable tokens representing paid access rights, or (d) creating hierarchical token markets mapped to URL path structures. The granularity of pricing is limited by traditional payment rails: there is no mechanism for a user to pay a fraction of a penny for a single article, a single API call, or a single data query.

3. **Lack of Payment Signalling Convention** — There is no established convention by which a URL can signal to a client — whether human or machine — that the resource requires payment before the resource is requested. Clients must make a request and receive a rejection before learning that payment is needed, and the rejection provides no standardised information about price, payment method, or token requirements.

4. **Machine-to-Machine Payment Inability** — As AI agents increasingly interact with web APIs on behalf of users, there is no standardised mechanism for an autonomous software agent to discover a price, make a payment, and receive access without human intervention. Existing payment systems require human authentication steps that are incompatible with autonomous operation.

5. **HTTP 403 — Authorisation Without Programmability** — When a server returns 403 Forbidden, the denial is typically opaque. The client knows it is forbidden but not why, nor what conditions would lift the prohibition. There is no standardised mechanism for: (a) encoding the specific conditions that must be satisfied, (b) enabling the client to programmatically satisfy those conditions, (c) creating auditable records of condition evaluation, or (d) composing conditions across multiple independent services.

6. **Fragmentation Across Layers** — Identity (401), payment (402), and authorisation (403) are treated as entirely separate concerns, implemented by different systems with no shared

protocol or data model. A user who has authenticated (resolved their 401) and paid (resolved their 402) may still be denied access (receive a 403) with no shared context between the three resolutions.

7. **No Native Tokenisation** — Web resources lack native economic primitives. Content cannot be natively priced, access rights cannot be natively traded, and identity cannot natively accumulate value. Existing tokenisation efforts (ERC-20, NFTs) operate on separate blockchain networks with no integration into the HTTP request-response cycle.

## Prior Art Limitations

Web monetisation proposals (W3C Web Monetization API) provide streaming payments but require browser extensions and do not use standard HTTP status codes. Payment channel networks (Bitcoin Lightning Network) reduce transaction costs but operate outside the HTTP protocol layer and require separate connection establishment. Recent implementations of the x402 protocol by Coinbase (published May 2025) address payments in isolation but do not integrate identity or permission layers, do not define URL path conventions for payment signalling, do not establish unilateral contract formation through HTTP responses, and operate exclusively on account-based blockchains (Ethereum/Base) using stablecoin transfers rather than native token economies with algorithmic pricing. OAuth 2.0 and OpenID Connect provide identity but not portability or composability across providers. Security token standards (ERC-1400, ERC-3643) provide compliance mechanisms but are not integrated with HTTP status codes or web-native discovery protocols. Proof of Work (Bitcoin, Ethereum pre-merge) provides consensus but performs computationally wasteful work unrelated to any useful indexing or cataloguing task.

No existing system combines HTTP status code mapping across identity, payment, and permission functions; a URL path convention for human-readable payment signalling; blockchain-native token economies with algorithmic pricing; DNS-based facilitator routing; and unilateral contract formation through standard HTTP responses in a unified protocol.

## Summary of the Invention

The present invention provides an integrated protocol suite ("the Suite") comprising three blockchain token protocol layers mapped to HTTP status codes:

a. **The $401 Identity Layer (HTTP 401 Unauthorized)** — A blockchain-native identity system comprising: a root inscription establishing the user's on-chain anchor and payout address; a plurality of strand inscriptions, each cryptographically linked to a verified OAuth identity provider (GitHub, Google, LinkedIn, X/Twitter, Microsoft, HandCash, domain ownership, or other providers); a type-gated identity strength classification that determines the user's trust level based on the categories of identity proof provided rather than the quantity; and public resolution endpoints enabling any party to verify a user's identity chain without contacting the original identity providers.

b. **The $402 Commerce Layer (HTTP 402 Payment Required)** — A content tokenisation system comprising: a dollar-prefixed URL path convention ("$address") wherein each URL path segment prefixed with a dollar sign constitutes an independent token market; HTTP 402 response headers encoding machine-readable pricing terms, payment addresses, and token market references; a configurable algorithmic pricing engine wherein the issuer selects a pricing model and parameters appropriate to their content; a hierarchical token tree model wherein revenue flows upward from child paths to parent paths; a Hash-to-Mint (HTM) utility token earned through Proof of Indexing — a consensus mechanism binding useful data-indexing work to proof-of-work mining challenges; and DNS-based facilitator routing enabling any domain to participate without code changes.

c. **The $403 Securities Layer (HTTP 403 Forbidden)** — A programmable access control and securities token system comprising: a condition engine evaluating configurable conditions (KYC level, jurisdiction, accreditation status, holding period, maximum holder count, transfer lock) against each proposed operation; a hierarchical access level system (public, basic, accredited, institutional, admin) with numeric rank comparison; an identity bridge delegating KYC evaluation to the $401 identity layer; an auditable compliance record for every condition evaluation; and transfer validation returning per-condition machine-readable results.

The three layers form an integrated stack: the $401 layer provides identity that the $402 layer references for revenue routing and the $403 layer requires for compliance gating. The $402 layer provides the economic settlement that the $403 layer may condition upon. The $403 layer provides the programmable access control that gates both $402 content and the issuance of securities tokens.

## Detailed Description of the Invention

### 1. System Architecture — The Tripartite Protocol Stack

The Suite maps three HTTP status codes to three blockchain token layers operating on the Bitcoin SV (BSV) blockchain. Each layer has a distinct function, a distinct token type, and a distinct on-chain inscription format, yet all three share a common inscription protocol prefix scheme and cross-reference one another's on-chain records.

| HTTP Code | Protocol Layer | Function | Token Type | Inscription Prefix |
|---|---|---|---|---|
| 401 | $401 Identity | Authentication | Identity token | `p: "401"` |
| 402 | $402 Commerce | Payment | Content/utility token | `p: "$402"` |
| 403 | $403 Securities | Authorisation | Security token | `p: "$403"` |

The layers are composable: a single HTTP request may trigger evaluation of all three layers in sequence. A server receiving a request first checks the client's $401 identity (authentication), then evaluates $402 pricing (payment), then checks $403 conditions (authorisation). Each layer's resolution produces on-chain records that subsequent layers may reference.

### 2. The $401 Identity Layer

#### 2.1 Root Inscription

When a user first registers, a root identity inscription is created on the BSV blockchain. The root inscription is the on-chain anchor for the user's entire identity. It contains:

```
{
  "p": "401",
  "op": "root",
  "v": "1.0",
  "payTo": "<BSV address for revenue>",
  "ts": "<ISO 8601 timestamp>"
}
```

The `payTo` field is significant: it establishes the address to which all $402 content revenue for this identity flows. This creates a direct, on-chain link between identity and economic participation.

The root inscription is recorded on-chain using one of two methods: (a) an OP_FALSE OP_RETURN data carrier output containing the protocol prefix "401", the content type "application/json", and the JSON payload; or (b) an Ordinals-compatible inscription envelope.

#### 2.2 Strand Inscriptions

For each identity provider the user connects, a strand inscription is created referencing the root:

```
{
  "p": "401",
  "op": "strand",
  "v": "1.0",
  "root": "<root inscription transaction identifier>",
  "provider": "<provider name>",
  "handle": "<user handle on provider>",
  "proofHash": "<SHA-256 hash of OAuth access token>",
  "metadata": { "<provider-specific public data>" },
  "ts": "<ISO 8601 timestamp>"
}
```

The `proofHash` field is a critical privacy mechanism: the user's OAuth access token is hashed using SHA-256 immediately upon receipt during the OAuth callback, and the raw token is discarded. Only the hash is inscribed on-chain. This proves that the user completed OAuth verification with the named provider at the stated time, without exposing the token itself.

### 2.3 Type-Gated Identity Strength

The invention introduces a type-gated identity classification that determines trust level based on the categories of identity proof provided, rather than the quantity. This is a deliberate anti-gaming mechanism: accumulating many OAuth accounts of the same type does not increase trust level. The classification operates as follows:

- **Level 1 (Basic)**: Any single OAuth strand. Achievable through automation; lowest trust.
- **Level 2 (Verified)**: Requires at least one strand of type: self-attestation, identity document (passport, driving licence, proof of address), camera verification, or video verification. Requires human participation.
- **Level 3 (Strong)**: Requires at least one strand of type: paid signing (economic commitment) or peer attestation (co-signature by another verified identity). Requires real-world economic or social commitment.
- **Level 4 (Sovereign)**: Requires at least one strand of type: KYC verification through a biometric verification service. Requires government-issued identity confirmation.

This classification ensures that identity strength cannot be inflated by creating many accounts of the same type. Fifty GitHub accounts still yield Level 1. One paid signing yields Level 3.

### 2.4 Additional Identity Operations

The protocol supports four additional on-chain operations:

- **Update**: Changes the payout address ( `payTo` field), enabling revenue redirection without creating a new identity.
- **Rotate**: Delegates identity authority to a new cryptographic key pair, enabling key migration without losing identity history.

- **Revoke**: Permanently invalidates the entire identity chain, with a mandatory reason field.
- **Service**: Registers a signing service as a participant in the identity ecosystem (detailed in Section 7).

### 2.5 Domain Verification

Domains may be cryptographically linked to $401 identities via DNS TXT records:

```
_401._domainkey.example.com TXT "v=401; p=<compressed public key>; t=<$401 root transacti
```

This enables any HTTP server to advertise its $401 identity through existing DNS infrastructure.

### 2.6 HTTP Authentication Integration

When a server requires $401 identity verification, it returns:

```
HTTP/1.1 401 Unauthorized
X-401-Min-Level: 2
X-401-Required-Strands: self_attestation
X-401-Verify-Url: https://path401.com/verify
```

The client responds with a signed proof header:

```
X-401-Token: <pubkey>|<scheme>|<timestamp>|<requestPath>|<$401-txid>|<signature>
```

The server verifies the signature, confirms the $401 transaction identifier references a valid root inscription with sufficient identity strength, and grants access.

### 2.7 Agent Delegation

$401 tokens support registration of autonomous AI agents as sub-identities. An agent registration includes: the agent's public key, capabilities, service endpoints (MCP, A2A), capital spending limits, and a cryptographic signature from the parent $401 identity. Agent trust inherits from the parent identity but operates under defined constraints.

## 3. The $402 Commerce Layer

### 3.1 The $Address Convention

The invention introduces a dollar-prefixed URL path convention ("$address") wherein each path segment prefixed with a dollar sign constitutes an independent economic entity:

```
$example.com                    → Site-level entity
$example.com/$blog              → Section-level entity
$example.com/$blog/$article     → Content-level entity
```

Each `$address` is a potential token market with its own supply, pricing, and holder base. The dollar sign is a legal sub-delimiter character under RFC 3986 (Uniform Resource Identifier: Generic Syntax) and RFC 1738, and is explicitly permitted in URL paths alongside characters such as hyphen, underscore, period, and tilde. The dollar prefix is syntactically distinguishable from standard URL paths, enabling middleware to intercept and process token-related requests without ambiguity. It also functions as a human-readable visual signal: the dollar sign is universally recognised as a currency symbol, providing an immediate indication that the resource has an associated cost. Client software, including AI agents, can inspect a URL prior to making a request and determine from the presence of the "$" character that payment will be required, enabling pre-emptive price discovery via the well-known endpoint.

### 3.2 Hierarchical Token Tree

The $address paths form a tree structure with the following properties:

1. Each path segment is an independent market with its own token supply and pricing.
2. Revenue flows upward: a configurable percentage of child path revenue (default 50%) flows to the parent path.
3. Child tokens do not dilute parent tokens; minting a child creates a new market, not new supply of the parent.
4. Holding a root path token provides exposure to all child path revenue, functioning as an index fund.
5. The tree structure provides natural curation: paths that attract economic activity (token purchases) signal value.
6. Early discovery is rewarded: first buyers of any path receive tokens at the lowest price on the curve.

### 3.3 HTTP 402 Response Protocol

When a client requests a $402-gated resource, the server returns:

```
HTTP/1.1 402 Payment Required
X-Protocol: $402
X-Price: <price in smallest unit>
X-Currency: SAT
```

```
X-Payment-Address: <BSV address or handle>
X-Token-Address: <$address of the token market>
X-Payment-Endpoint: <URL for payment submission>
```

The response body contains a machine-readable JSON document specifying: the $address, pricing model and parameters, current supply, current price, payment address, and API endpoints for resolution, payment, and verification.

This 402 response constitutes a standing offer under contract law. When the client makes payment to the specified address, acceptance occurs and a unilateral contract forms. The server verifies payment and delivers the content, and the client receives transferable tokens representing their access rights.

### 3.4 Configurable Algorithmic Pricing

The $402 layer provides a pricing engine that supports multiple algorithmic pricing models, the selection and parameterisation of which are at the sole discretion of the token issuer. The system architecture separates the pricing engine (which computes prices) from the protocol layer (which communicates prices via HTTP headers), such that any pricing model may be plugged into the engine without modifying the protocol.

In a preferred embodiment, the system provides a library of pricing models including but not limited to: ascending bonding curves, descending decay curves, fixed pricing, and linear models. The issuer selects a model and configures its parameters (base price, calibration constants, ceiling valuation) at the time of token creation. The pricing engine computes the current price dynamically based on the current state of the token market (supply minted, treasury balance, time elapsed, or other inputs as defined by the model).

The pricing model is fully transparent and computable by any client. No negotiation, auction, or human pricing decision is required. The price at any moment is a pure function of the publicly observable on-chain state, and the pricing function is declared in the discovery document. Any client can independently verify the current price by reading the on-chain state and applying the declared pricing function.

This mechanism ensures that pricing is deterministic, auditable, and free from centralised control. The content creator selects the pricing function at token creation; thereafter, the price is governed entirely by the algorithm and the observable state. Critically, the protocol does not mandate any specific pricing algorithm — the choice of model and parameters is at the sole discretion of the token issuer. The protocol's contribution is the architecture that communicates computed prices to clients via HTTP headers and discovery endpoints, enabling programmatic payment regardless of which pricing model the issuer has selected.

### 3.5 Bearer Token Access Model

Tokens acquired through $402 payment function as bearer instruments: holding the token confers ongoing access rights to the gated content, and the token may be transferred to other parties on secondary markets. Token holders may also receive proportional dividends from revenue generated by subsequent purchasers, creating an economic incentive for early participation.

### 3.6 Hash-to-Mint (HTM) Utility Token and Proof of Indexing

The $402 layer includes a utility token (the "$402 token") with a fixed maximum supply, earned exclusively through a novel consensus mechanism called "Proof of Indexing."

**Proof of Indexing** replaces wasteful hash computation with productive data-indexing work. The mechanism operates as follows:

1. **Work Performance**: Participating nodes perform verifiable indexing tasks: validating token transfers, serving gated content to paying users, verifying ticket stamp chains, updating market data, and relaying peer-to-peer messages.

2. **Work Commitment**: Completed work items are assigned identifiers. The identifiers are sorted and combined into a SHA-256 Merkle root (the "work commitment"). This work commitment is a 32-byte value binding the useful work performed to the mining challenge.

3. **Mining Challenge**: The mining challenge is constructed as: `challenge = previousTransactionId || workCommitment || destinationAddress || nonce`. The miner iterates nonces until `SHA256(SHA256(challenge))` produces a hash value below the current difficulty target.

4. **On-Chain Verification**: The valid solution is submitted to a smart contract on the BSV blockchain. The smart contract verifies: (a) the work commitment is exactly 32 bytes; (b) the double-SHA256 hash of the challenge is below the target; (c) the supply has not been exhausted. Upon verification, tokens are minted to the destination address.

5. **Difficulty Adjustment**: Difficulty adjusts every 144 solutions using a Bitcoin-style algorithm: `new_target = old_target * (actual_time / expected_time)`, clamped to a maximum 4x change per adjustment period. The target solution time is 10 minutes.

6. **Halving Schedule**: The token reward per solution halves at fixed intervals, mirroring Bitcoin's emission schedule. Initial reward: 50 tokens per solution. The halving schedule produces a predictable, declining emission curve over 33 halving eras.

The work commitment mechanism is the key innovation: it cryptographically binds productive indexing work to the proof-of-work solution. The Merkle root is verifiable by peers via the gossip network — they may request individual work items and verify the Merkle tree. This ensures miners cannot earn tokens without performing genuine indexing work.

### 3.7 Ticket Stamp Chains

Content access creates a cryptographic provenance trail called a "stamp chain." Each access event appends a stamp comprising: a sequence number, the indexer's public key, the owner's address, the action performed (mint, transfer, serve, verify), and a cryptographic signature. The complete stamp chain provides an auditable record of every interaction with the content, from creation through all transfers and accesses.

### 3.8 DNS-Based Facilitator Routing (X Protocol)

The invention introduces a DNS-based routing mechanism enabling any domain to participate in the protocol suite without code changes, using three standardised DNS CNAME records:

```
x401.example.com.    CNAME    path401.com.    ; Identity layer
x402.example.com.    CNAME    path402.com.    ; Payment layer
x403.example.com.    CNAME    path403.com.    ; Conditions layer
```

This is analogous to how MX records route email: the domain owner adds DNS records, and the protocol infrastructure handles the rest. A DNS TXT record at `_x-protocol.example.com` announces which layers are active.

Domain ownership is verified through a triple-proof mechanism: (a) a DNS TXT record at `_path402.<domain>` containing a verification code; (b) an HTTP well-known endpoint at `/.well-known/path402.json` containing the same code; and (c) an on-chain signature inscription.

### 3.9 Discovery Protocol

The system provides standardised discovery endpoints:

- `/.well-known/$402.json` — Returns protocol version, supported extensions, root token details, and child token markets with current pricing.
- `/.well-known/x402.json` — Returns facilitator details, supported payment networks, fee structure, and API endpoints.

These endpoints enable automated discovery by AI agents and programmatic clients, following the established `.well-known` convention (RFC 8615).

### 3.10 Peer-to-Peer Gossip Network

The $402 layer maintains a libp2p-based gossip network for propagating token market state. The network uses Noise encryption, Yamux multiplexing, and GossipSub message propagation across topic-based channels including: token announcements, transfer events, stamp chain events, content negotiation, block announcements, and transaction relay.

## 4. The $403 Securities Layer

### 4.1 Security Token Registration

The $403 layer enables registration of securities tokens with configurable compliance conditions. A security token registration comprises: a unique identifier, ticker symbol, name, security type (equity, debt, fund, derivative, or utility), issuer address, issuer $401 identity root reference, total and maximum supply, decimal precision, transfer restriction mode, issuance status, jurisdiction, and required KYC level.

The `issuer_identity_root` field creates a mandatory cryptographic link between the security token and the issuer's $401 on-chain identity. Securities cannot be issued anonymously — the issuer must have a verified $401 identity of sufficient strength.

### 4.2 Programmable Condition Engine

The $403 layer provides a condition engine that evaluates configurable conditions against each proposed operation. Six condition types are defined:

1. **KYC Level** — Minimum $401 identity strength level required. Evaluation is delegated to the $401 identity layer via a configured identity bridge, maintaining separation of concerns between identity verification and securities compliance.
2. **Jurisdiction** — Geographic restriction based on ISO 3166-1 country codes. The condition specifies permitted or prohibited jurisdictions.
3. **Accreditation** — Investor accreditation status as defined by applicable securities regulation.
4. **Holding Period** — Minimum duration tokens must be held before transfer is permitted.
5. **Maximum Holders** — Cap on the total number of distinct addresses that may hold the token simultaneously.
6. **Transfer Lock** — Absolute prohibition on transfers, regardless of other conditions.

Conditions are attached to specific security tokens and may be added, modified, or deactivated by the issuer. Each condition evaluation produces a machine-readable result specifying which condition was evaluated, whether it was satisfied, and a detailed reason.

### 4.3 Hierarchical Access Levels

The $403 layer defines five hierarchical access levels: public (0), basic (1), accredited (2), institutional (3), and admin (4). Each level inherits all rights of lower levels. Access is granted per-security per-address, with optional expiry dates. The access grant records: the security token, grantee address, grantee $401 identity root, access level, grant status, granting authority, timestamp, expiry, and which conditions were satisfied.

### 4.4 Identity Bridge — Cross-Protocol KYC Delegation

When the $403 condition engine evaluates a KYC-level condition, it delegates to the $401 identity layer via a configured bridge URL. The bridge queries the $401 system to determine the subject's identity strength level, strand types, and verification timestamps. This delegation maintains separation of concerns: the $401 layer is the authoritative source of identity, and the $403 layer is the authoritative evaluator of conditions. Neither layer stores the other's data.

**4.5 Transfer Validation**

Every proposed token transfer is validated against all active conditions for the security. The transfer validation returns a structured result comprising: (a) whether the transfer is approved; (b) a unique transfer identifier; (c) a human-readable reason for denial (if denied); and (d) a per-condition breakdown showing each condition type, whether it was satisfied, and a detailed explanation.

This per-condition breakdown is a significant innovation over existing compliance systems, which typically return only a binary approve/deny. The granular result enables clients to display precisely which conditions remain unsatisfied and what the user must do to satisfy them — transforming the opaque HTTP 403 "Forbidden" into a navigable set of steps.

**4.6 Compliance Audit Trail**

Every condition evaluation creates an immutable compliance record comprising: the security token, the subject's address and $401 identity root, the operation attempted, the compliance status (pending, approved, rejected, suspended, expired), the checker's identity, the check type, detailed results, and an optional expiry. Compliance approvals are time-limited, requiring periodic re-evaluation.

**4.7 Proof of Indexing for Compliance**

The $403 layer includes its own Proof of Indexing mining mechanism wherein participating nodes earn $403 tokens by performing verifiable compliance indexing work: validating security token transactions, evaluating compliance conditions, and indexing audit records. The mining mechanism follows the same proof-of-work-bound-to-useful-work pattern as the $402 layer, with its own independent difficulty target and halving schedule.

## 5. Cross-Layer Integration

### 5.1 The Three-Layer Resolution Sequence

When a server receives a request for a resource gated by all three layers, the resolution proceeds as follows:

1. **$401 Check**: The server examines the request for a $401 identity proof (X-401-Token header). If absent or insufficient, the server returns HTTP 401 with X-401 headers indicating the required identity level and strand types.
2. **$402 Check**: If identity is satisfied, the server checks whether the resource requires payment. If so and no payment proof is present, the server returns HTTP 402 with pricing headers, payment address, and token market reference.
3. **$403 Check**: If both identity and payment are satisfied, the server evaluates $403 conditions (jurisdiction, accreditation, holding period, etc.). If conditions are not met, the server returns HTTP 403 with a structured response indicating which conditions are unsatisfied and what is required.
4. **Content Delivery**: If all three layers are satisfied, the content is delivered with a 200 OK response.

This sequence mirrors the HTTP status code semantics: 401 (who are you?), 402 (will you pay?), 403 (are you permitted?), 200 (here is your content).

### 5.2 On-Chain Cross-References

Each layer's on-chain inscriptions may reference inscriptions from other layers:

- $402 content token inscriptions reference the issuer's $401 identity root, establishing who created the content.
- $403 security token inscriptions reference the issuer's $401 identity root, establishing who issued the security.
- $403 compliance records reference the subject's $401 identity root, linking compliance evaluations to verified identities.
- $403 condition evaluations may reference $402 token holdings, enabling conditions such as "must hold N tokens of $address X."

### 5.3 Shared Gossip Infrastructure

All three layers may share a common peer-to-peer gossip network infrastructure, with layer-specific topic channels. A single node may participate in all three layers' gossip networks, earning tokens from each layer based on the indexing work it performs — a hybrid mining model with three independent reward streams from one daemon process.

### 6. Autonomous AI Agent Integration

#### 6.1 MCP (Model Context Protocol) Tools

Each layer exposes a set of tools via the Model Context Protocol, enabling AI agents to: resolve identities ($401), discover and pay for content ($402), check compliance and request access grants ($403). AI agents can autonomously navigate the three-layer resolution sequence, discovering pricing via 402 responses, constructing payments, and retrying with proof.

#### 6.2 Agent Chaining

The $402 layer supports autonomous multi-step agent pipelines wherein an AI agent: (a) discovers other $402-compatible agents via `/.well-known/x402-info`; (b) plans a multi-step pipeline from a natural language prompt; (c) executes each step, paying each agent via the $402 protocol; (d) pipes outputs between steps; and (e) budget-constrains the entire chain.

### 7. Federated Signing

The $401 layer supports a federated signing protocol wherein multiple independent signing services (e.g., different document-signing platforms) can interoperate as peers. Each service registers on-chain with an `op: "service"` inscription and exposes a `/.well-known/bit-sign.json` discovery endpoint. Documents signed on one platform can be verified, co-signed, or counter-signed on another, with all signatures referencing the signatories' $401 identity chains.

---

## Brief Description of Drawings

The following drawings would accompany this application:

- **Figure 1** — System architecture diagram showing the tripartite protocol stack ($401, $402, $403) with cross-layer references and shared gossip infrastructure.
- **Figure 2** — HTTP resolution sequence diagram showing the 401 → 402 → 403 → 200 cascade for a fully-gated resource.
- **Figure 3** — $401 identity chain structure showing root inscription and strand inscriptions for each OAuth provider, with type-gated strength classification.
- **Figure 4** — $Address hierarchical token tree showing path-based token markets with upward revenue flow.
- **Figure 5** — Proof of Indexing mining flow showing work performance, Merkle root computation, mining challenge construction, and on-chain verification.
- **Figure 6** — $403 condition evaluation engine showing the six condition types, the identity bridge to $401, and the per-condition result structure.
- **Figure 7** — DNS-based facilitator routing showing CNAME records, discovery endpoints, and the X Protocol layer.

- **Figure 8** — Transfer validation flow showing condition-by-condition evaluation with machine-readable results.
- **Figure 9** — Cross-layer data flow showing how $401 identity roots are referenced by $402 content tokens and $403 security tokens.
- **Figure 10** — Autonomous AI agent pipeline showing discovery, payment, and content delivery across multiple $402-compatible services.

## Initial Claims

*Note: These claims are provided in sketch form for the purposes of establishing a priority date. Formal claims will be drafted and filed within 12 months in accordance with UKIPO rules.*

### Claim 1 — HTTP Status Code Token Protocol Suite

A system for tokenising web resources on a blockchain, the system comprising three integrated protocol layers: (a) a first layer mapped to HTTP status code 401 (Unauthorized), comprising an on-chain identity token system wherein a root inscription establishes a user's blockchain anchor and a plurality of strand inscriptions, each cryptographically linked to a distinct verified identity provider, form an identity chain with a type-gated strength classification; (b) a second layer mapped to HTTP status code 402 (Payment Required), comprising a content commerce token system wherein dollar-prefixed URL path segments each constitute an independent token market with configurable algorithmic pricing, and wherein HTTP 402 responses encode machine-readable pricing terms enabling programmatic payment and token acquisition; (c) a third layer mapped to HTTP status code 403 (Forbidden), comprising a programmable access control and securities token system wherein configurable conditions are evaluated against proposed operations and evaluation results are returned in per-condition machine-readable form; wherein the three layers are integrated such that the second and third layers reference the first layer's identity inscriptions, and the third layer may condition access upon holdings in the second layer's token markets.

### Claim 2 — Three-Layer HTTP Resolution Method

A method of resolving access to a web resource using blockchain token protocols, the method comprising: (a) receiving an HTTP request for a resource; (b) checking for a blockchain-based identity proof corresponding to HTTP 401 Unauthorized and, if absent or insufficient, returning an HTTP 401 response specifying the required identity strength and strand types; (c) if identity is satisfied, checking for a payment proof corresponding to HTTP 402 Payment Required and, if absent, returning an HTTP 402 response encoding the resource price, payment address, and token market reference; (d) if payment is satisfied, evaluating programmable conditions corresponding to HTTP 403 Forbidden and, if conditions are unsatisfied, returning an HTTP 403 response specifying each unsatisfied condition and the steps required to satisfy it; (e) if all three layers are satisfied, delivering the resource with an HTTP 200 response.

### Claim 3 — Dollar-Prefixed URL Path Token Markets

A system for creating hierarchical token markets from URL path structures, the system comprising: (a) a convention wherein URL path segments prefixed with a dollar sign each constitute an independent token market on a blockchain; (b) a hierarchical tree structure wherein each path segment is a node with its own token supply and pricing; (c) a revenue flow mechanism wherein a configurable proportion of revenue from child path tokens flows upward to parent path tokens; (d) an HTTP middleware component that intercepts requests to dollar-prefixed paths and serves token market data or initiates the payment

protocol; wherein holding a token at any node in the tree confers proportional economic rights to revenue generated by that node and its descendants.

### Claim 4 — Type-Gated Identity Strength Classification

A method of classifying the strength of a blockchain-based digital identity, the method comprising: (a) maintaining an identity chain comprising a root inscription and a plurality of strand inscriptions on a blockchain, each strand inscription linked to a distinct identity verification method; (b) categorising each strand by verification type rather than by provider; (c) determining the identity strength level based on the highest-assurance category of strand present, regardless of the total number of strands; wherein accumulating multiple strands of the same category does not increase the identity strength level, thereby providing an anti-gaming mechanism against Sybil attacks.

### Claim 5 — Proof of Indexing Consensus Mechanism

A consensus mechanism for minting blockchain tokens, the mechanism comprising: (a) performing verifiable data-indexing work and assigning an identifier to each completed work item; (b) computing a cryptographic Merkle root from the sorted identifiers of the completed work items, the Merkle root constituting a "work commitment"; (c) constructing a mining challenge comprising the work commitment and iterating nonces until a double-SHA256 hash of the challenge falls below a difficulty target; (d) submitting the valid solution to a smart contract on a blockchain, the smart contract verifying the work commitment length, the hash against the target, and the remaining token supply; (e) minting tokens to the miner's address upon successful verification; wherein the work commitment cryptographically binds productive indexing work to the proof-of-work solution such that tokens cannot be earned without performing genuine data-indexing tasks.

### Claim 6 — Programmable HTTP 403 Condition Engine

A system for transforming HTTP 403 (Forbidden) responses from opaque denials into navigable condition sets, the system comprising: (a) a condition engine configured to evaluate a plurality of configurable conditions against a proposed operation on a security token; (b) an identity bridge that delegates KYC-level condition evaluation to a separate blockchain-based identity protocol layer; (c) a structured response format that, for each condition, specifies: the condition type, whether it was satisfied, and a machine-readable explanation; wherein the client receiving the 403 response can programmatically determine precisely which conditions remain unsatisfied and what actions are required to satisfy them.

### Claim 7 — DNS-Based Protocol Layer Routing

A method of adding blockchain-based identity, payment, and access control capabilities to a web domain using DNS records, the method comprising: (a) adding one or more CNAME DNS records pointing standardised subdomains to protocol facilitator services; (b) optionally adding a DNS TXT record announcing which protocol layers are active; (c) the protocol facilitator services providing identity verification, payment processing, and condition evaluation on behalf of the domain; wherein the domain

owner requires no code changes, SDK integration, or server-side implementation to participate in the protocol suite, analogous to how MX records add email capability to a domain.

### Claim 8 — Cross-Protocol Condition Composition

A method of evaluating access conditions for a blockchain security token, wherein at least one condition references the state of a different protocol layer, the method comprising: (a) receiving a proposed operation on a security token in a third protocol layer; (b) evaluating a first condition by querying a first protocol layer to determine the subject's identity strength; (c) evaluating a second condition by querying a second protocol layer to determine the subject's token holdings; (d) combining the results of all condition evaluations into a unified approval or denial with per-condition detail; wherein conditions may reference state across independent protocol layers, enabling composite access rules spanning identity, payment, and permission.

### Claim 9 — OAuth Proof Hash as On-Chain Identity Attestation

A method of creating a blockchain-based attestation of identity provider verification, the method comprising: (a) completing an OAuth authentication flow with an identity provider and receiving an access token; (b) immediately computing a SHA-256 hash of the access token; (c) discarding the raw access token; (d) inscribing on a blockchain a strand record comprising: the identity provider name, the user's handle on that provider, the computed hash, and a reference to the user's root identity inscription; wherein the inscription proves that the user completed OAuth verification with the named provider without storing or exposing the access token, and third parties can verify the inscription's authenticity from the blockchain alone.

### Claim 10 — Unilateral Contract Formation via HTTP 402

A method of forming a unilateral contract for digital content access, the method comprising: (a) a server returning an HTTP 402 Payment Required response encoding specific terms comprising: a price, a payment address, a currency denomination, and a token market reference; (b) a client making payment to the specified address in accordance with the encoded terms; (c) the server verifying the payment on the blockchain; (d) the server delivering the content and minting a transferable token to the client's address; wherein the HTTP 402 response constitutes a standing offer, the client's payment constitutes acceptance, and the minted token represents a transferable bearer instrument conferring ongoing access rights to the purchased content.

## Abstract

An integrated protocol suite mapping three HTTP status codes to three blockchain token layers on the Bitcoin SV network. HTTP 401 (Unauthorized) maps to an identity layer comprising on-chain root and strand inscriptions linked to verified OAuth identity providers, with a type-gated strength classification resistant to Sybil attacks. HTTP 402 (Payment Required) maps to a commerce layer comprising dollar-prefixed URL path token markets with hierarchical revenue flow, configurable algorithmic pricing, and a Hash-to-Mint utility token earned through Proof of Indexing — a novel consensus mechanism binding productive data-indexing work to proof-of-work mining. HTTP 403 (Forbidden) maps to a securities layer comprising a programmable condition engine with six condition types, cross-protocol identity bridging, and per-condition machine-readable results. The three layers form an integrated stack: identity gates payment, identity and payment gate securities, and all layers share a common peer-to-peer gossip infrastructure. DNS-based facilitator routing enables any domain to participate without code changes. The suite transforms opaque HTTP error responses into navigable, programmable, token-based resolution paths.

---

Document prepared for UKIPO filing. Priority date to be established upon submission.
Applicant: The Bitcoin Corporation Ltd
Inventor: Richard Boase
Date of preparation: 28 February 2026