

\$402 Protocol Specification

Tokenized Attention Economy with Social Scaling

Version: 3.0.0 **Status:** Living Document **Reference Implementation:**
[PATH402.com](https://path402.com)

Overview

\$402 is a protocol for tokenized attention markets. Every participant mints their own token, creating a market for their time and content. The protocol harnesses viral memecoin dynamics to bootstrap the creator economy—every domain becomes a shareholder company, every path becomes a tradable asset.

The Progression:

1. **Content Tokenization** (v1-v2): Turn URL paths into shareholder businesses
2. **Personal Tokenization** (v3): Turn individuals into attention markets

Core Principles

1. **Everyone has a token** - Your token is your attention market
2. **Fixed supply** - 1 billion tokens per person (no minting/burning)
3. **Time-based access** - Tokens purchase connection time (1 token = 1 second default)
4. **Creator-controlled float** - You decide how many tokens to sell
5. **Social bootstrapping** - Friends invest in friends

Token Economics

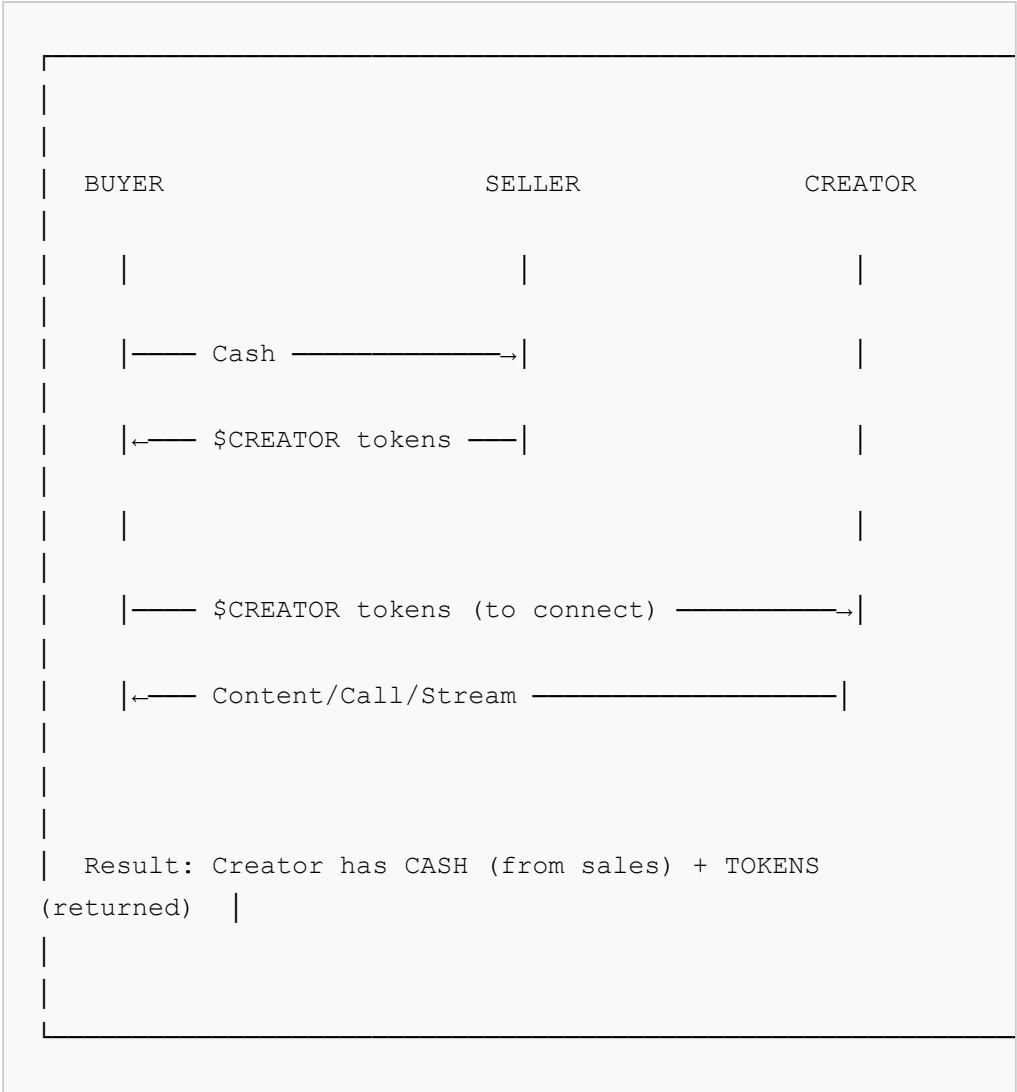
Supply Model

Per-person supply: 1,000,000,000 tokens (1 billion)
No minting after genesis
No burning - tokens circulate

Access Pricing

Base rate: 1 token = 1 second of connection
Configurable per creator: 1-100 tokens/second
Multicast: Same rate, split across viewers (or flat per viewer)

The Economic Flow



Creator Float Control

Strategy	Token Float	Price	Result
Exclusive	Low (hold most tokens)	High	Premium access, fewer connections
Accessible	High (sell freely)	Low	Mass audience, lower per-connection value
Balanced	Medium	Market-driven	Organic price discovery

Social Scaling

The Friend Investment Model

Users naturally invest in people they care about:

```
I value my friend
      ↓
I buy MORE tokens than I need
      ↓
I stake tokens (earn from their success)
      ↓
I complete KYC (because I trust them)
      ↓
They succeed → I profit + maintain access
      ↓
Network grows through real relationships
```

Why This Works

1. **Aligned incentives** - Supporting friends = potential profit
 2. **Natural liquidity** - Social circles create baseline markets
 3. **Trust bootstrapping** - KYC happens where trust exists
 4. **Anti-spam** - Connecting costs tokens (economic friction)
 5. **No platform rent** - Direct peer-to-peer settlement
-

Staking & Dividends

Staking Mechanism

```
Stake $CREATOR tokens
    ↓
Receive share of creator's revenue
    ↓
Revenue = all token purchases for their content
    ↓
Dividend = (your_stake / total_staked) × revenue ×
dividend_rate
```

Dividend Claims & KYC

- **Basic access:** No KYC required (permissionless)
- **Dividend claims:** KYC required
- **Identity anchor:** Phone contract, government ID, or trusted verifier
- **Result:** Voluntary identity layer where money flows out

Revenue Split (Default)

```
Creator revenue from token sales:
├─ 70% → Creator wallet
├─ 20% → Staker dividend pool
└─ 10% → Protocol treasury ($402 token holders)
```

Connection Types

1. Direct Call (1:1)

```
Caller spends: N tokens (for N seconds)
Creator receives: N tokens
Bidirectional: Both parties spend each other's tokens
Net cost: Difference in token values
```

2. Multicast Stream (1:Many)

```
Creator streams content
Each viewer spends: N tokens/second
Creator receives: N × viewers tokens
Scales to unlimited viewers
```

3. Asymmetric Value Calls

```
$RICHARD worth: 1000 sats/token
$BOB worth: 100 sats/token

Bob calls Richard:
- Bob spends $RICHARD tokens (expensive)
- Richard spends $BOB tokens (cheap)
- Net: Bob pays ~900 sats/second to talk to Richard

Richard calls Bob:
- Same mechanics, reversed
- Net: Richard earns ~900 sats/second
```

Client Features

1. Identity & Minting

YOUR TOKEN: \$YOURNAME
Supply: 1,000,000,000
Float: 100,000,000 (10% for sale)
Floor price: 500 sats
Access rate: 1 token/second

2. Discovery & Indexing

DISCOVER

Search tokens...

TRENDING

FRIENDS

NEW

\$SPIELBERG

\$ROGAN

\$MRBEAST

\$ALICE

\$BOB

\$CAROL

\$JANE

\$MARK

\$SARA

3. Portfolio & Rankings

MY PORTFOLIO			
Token	Balance	Value	Staked
\$ALICE	50,000	\$25.00	Yes
\$BOB	10,000	\$5.00	No
\$CAROL	100,000	\$75.00	Yes
Total: \$105.00		Dividends: \$12.50	

4. Video/Audio Calls

VIDEO FEED	
Calling: \$ALICE	
Rate: 1 token/sec (500 sats/sec)	
Balance: 50,000 tokens (13.8 hours)	
[Mute] [Video] [End Call]	

5. Staking Panel

STAKING

\$ALICE - Staked: 25,000 tokens

└ Your share: 2.5%

└ Total pool: 1,000,000 staked

└ This month: \$50.00 revenue

└ Your dividend: \$1.25

[Stake More] [Unstake] [Claim]

6. KYC Tab

IDENTITY VERIFICATION

Status: ✓ Verified

Method: Phone contract

Verified: 2026-02-01

Linked tokens (can claim dividends):

└ \$ALICE ✓

└ \$BOB ✓

└ \$CAROL ✓

[Add Verification] [Manage]

Protocol Messages

Token Operations

```
interface TokenTransfer {
  token_id: string;      // e.g., "$RICHARD"
  from: string;          // sender address
  to: string;            // recipient address
  amount: number;        // tokens transferred
  purpose: 'purchase' | 'access' | 'stake' | 'unstake' |
  'dividend';
}
```

Connection Handshake

```
interface ConnectionRequest {
  caller_token: string;  // Caller's token ID
  callee_token: string;  // Callee's token ID
  caller_balance: number; // Tokens caller holds of callee
  requested_duration: number; // Seconds requested
  connection_type: 'call' | 'stream' | 'chat';
}

interface ConnectionAccept {
  session_id: string;
  rate: number;          // Tokens per second
  accepted_duration: number;
}
```

Staking Operations

```
interface StakeRequest {
  token_id: string;
  amount: number;
  staker: string;
}

interface DividendClaim {
  token_id: string;
  claimant: string;
  kyc_proof: string;      // Reference to KYC verification
  period: string;         // e.g., "2026-02"
}
```

BSV Implementation

Why BSV Only

Requirement	BSV	ETH/Others
1 token/second micropayments	✓ <0.001¢ fees	✗ \$0.50+ fees
Real-time settlement	✓ Instant	✗ Block times
Scalability	✓ Unbounded	✗ Gas limits
Token standard	BSV-20	ERC-20

On-Chain vs Off-Chain

On-chain (BSV):

- └ Token genesis (1bn supply)
- └ Large transfers
- └ Staking/unstaking
- └ Dividend distributions
- └ KYC attestations

Off-chain (Gossip):

- └ Connection handshakes
 - └ Micro-transfers during calls
 - └ Real-time balance updates
 - └ Periodic settlement to chain
-

Pricing Formula

Ticket Stamp Chains: The Indexer Solution

The Core Innovation: Tickets accumulate cryptographic stamps as they're validated and used, creating a trust layer that solves the indexer incentive problem through pure Bitcoin economics.

The Problem

When Harry buys a ticket to `fred.com/$video`:

1. Server must validate the ticket UTXO exists on-chain
2. Blockchain is large, validation is slow without indexing
3. If all payment goes to Fred, indexers have no incentive to index

The Paradox: We need indexers to make the system fast, but indexers need payment to exist.

The Solution: Stamp Chains

Every time a ticket is validated and used, the indexer adds a **stamp** to the ticket's chain. Payment is split between creator and indexer:

```
Ticket Price: 1000 sats
├ Creator (Fred): 950 sats (95%)
└ Indexer Fee: 50 sats (5%)
```

Stamp Chain Structure:

```
{
  "stampChain": [
    {
      "seq": 1,
      "indexer": "indexer-a.com",
      "owner": "harry_pubkey",
      "blockHeight": 837492,
      "feePaid": 50,
      "signature": "304402..."
    }
  ]
}
```

Trust Accumulation

Tickets gain value as stamp chains grow:

```
Fresh ticket (0 stamps):      1000 sats
After 100 stamps:              1300 sats (+30% trust premium)
After 1000 stamps:            2000 sats (+100% viral
premium)
```

Network Effects

1. **Economic Filtering:** Spam gets ignored (no fee potential), viral content attracts indexers

2. **Content Discovery:** Sort by stamp count = quality ranking
3. **Indexer Competition:** Popular content = higher revenue = better service
4. **Creator Incentives:** Viral content accumulates stamps faster, increasing secondary market value

Why This Works

- **No Bootstrap Token Needed:** Stamps ARE the proof of work
- **Pure Bitcoin Economics:** Fees flow naturally from users → creators + indexers
- **Self-Sustaining:** Market forces optimize service quality
- **Viral Dynamics:** Positive feedback loop between popularity and infrastructure

See: [STAMP_CHAIN_SPEC.md](#) for complete technical specification.

sqrt_decay Model

```
price = base_price / sqrt(supply_sold + 1)
```

- **Early buyers:** Lower price (reward early supporters)
- **Later buyers:** Higher price (scarcity premium)
- **Creator benefit:** Early friends get deals, mass market pays more

Example

```
Base price: 500 sats  
Supply sold: 0 → Price: 500 sats  
Supply sold: 99 → Price: 50 sats  
Supply sold: 9,999 → Price: 5 sats  
Supply sold: 999,999 → Price: 0.5 sats
```

Security Considerations

Anti-Spam

- Connection requires tokens (economic cost)
- No tokens = no access
- Spam is expensive

Privacy

- Basic access: Pseudonymous (just addresses)
- Dividend claims: KYC required (real identity)
- User controls identity disclosure level

Sybil Resistance

- Creating fake accounts costs nothing
 - But fake accounts have no social graph
 - No friends = no token buyers = worthless token
 - Social proof required for value
-

Legal Compliance & Liability

Corporate Register vs Token Ownership

The company has liability and must produce a confirmation statement each year showing an up-to-date register of members. However, Members can trade tokens that act as a claim on their shares without permission.

So while the confirmation statement may say Alice holds Title, she may have already sold her interest to Bob, who has already sold it to Charlie. The confirmation statement says Alice owns the share on the register, but that is still legally compliant, even though the reality is that Charlie holds the asset. As long as each individual pays their own taxes, they are compliant too.

Roadmap

Phase 1: Core Protocol ✓

- ☒ Token model defined
- ☒ Access mechanics specified
- ☒ Economic flows documented

Phase 2: Client Implementation

- ☐ Video/audio call module
- ☐ Portfolio tracking
- ☐ Discovery/search
- ☐ Staking UI
- ☐ KYC integration

Phase 3: Network Launch

- ☐ Bootstrap peers
- ☐ Initial token minting
- ☐ Creator onboarding
- ☐ Market maker liquidity

Phase 4: Ecosystem

- ☐ Mobile apps
 - ☐ Browser extension
 - ☐ API for third-party apps
 - ☐ Creator tools
-

Summary

\$402 creates an attention economy where:

1. **Everyone has a price** - Your token represents your time's market value
2. **Friends invest in friends** - Social relationships create liquidity
3. **Creators control access** - Your float, your rules
4. **Speculators align with supporters** - Staking rewards belief
5. **KYC is voluntary but incentivized** - Need it for dividends
6. **BSV enables micropayments** - 1 token/second is economically viable

The network scales through viral memecoin dynamics—combining speculation, ownership, and utility to bootstrap the creator economy.

Personal Token Minting

BSV21 Standard

Personal tokens use the BSV21 fungible token standard with \$402 extensions:

```
{
  "p": "bsv-21",
  "op": "deploy",
  "tick": "$RICHARD",
  "max": "1000000000",
  "dec": "0",
  "path402": {
    "accessRate": 1,
    "protocol": "path402",
    "version": "1.0.0"
  },
  "metadata": {
    "name": "RICHARD",
    "description": "Access token for Richard",
    "avatar": "https://...",
    "website": "https://..."
  }
}
```

Token Properties

Property	Value	Description
supply	1,000,000,000	Fixed, no further minting
decimals	0	Whole tokens only
accessRate	1-100	Tokens per second for connections
burning	Disabled	Tokens circulate, never destroyed

Proof of Serve

Nodes earn rewards through actual network contribution:

Action	Description	Reward Weight
serve	Deliver content to requesters	High
relay	Forward gossip messages	Medium
index	Maintain accurate indexes	Medium
validate	Verify transactions	Low

Reward Formula:

```
node_reward = (node_serves / total_network_serves) ×
daily_reward_pool
```

Why Not Proof of Work?

Proof of Work	Proof of Serve
Rewards hash computation	Rewards actual service
Wastes electricity	Uses real network work
Centralizes to ASICs	Scales with usage
One winner per block	Everyone who serves earns

Version: 3.0.0 Last Updated: 2026-02-05