

DigiRails: Payment Infrastructure for Autonomous AI Agents

February 2026

The Problem No One Has Solved

AI agents are learning to act. They book flights, write code, analyze data, manage infrastructure. Within months, they will need to buy and sell services from each other — inference, compute, data, verification — at machine speed, in volumes no human could manage.

There is no payment system built for this.

Credit cards require human account holders, minimum transaction amounts, and 2-3% fees that make sub-cent payments impossible. Bank transfers take days. PayPal needs an email address and a human behind it. These systems were designed for people buying things from companies. They break completely when the buyer is a Python script and the seller is a container running on the other side of the world.

Cryptocurrency was supposed to solve this. It hasn't — not yet. Bitcoin's 10-minute blocks and volatile fees make it impractical for micropayments. Lightning Network reintroduces the intermediaries it was designed to eliminate — well-capitalized hubs that route payments, hot wallets that must stay online, channels that lock capital. Ethereum's Layer 2 rollups depend on validators, bridges, and governance tokens — trust assumptions that autonomous software cannot evaluate. Solana and Sui offer speed but have demonstrated either untenable outages or the outright willingness to halt the chain and freeze assets when politically convenient.

The API key model — where a human pre-purchases credits from a provider using a credit card — works today but doesn't scale. An agent that needs satellite weather data from Provider A, legal analysis from Provider B, and code review from Provider C requires a human to set up and fund three separate accounts in advance. When the agent encounters Provider D at runtime — a specialized service it discovers on the fly — it's stuck. No account, no API key, no way to pay.

AI agents need payment infrastructure that is:

- **Instant.** Settlement in seconds, not minutes or days.
- **Cheap.** Sub-cent fees, so a \$0.001 API call doesn't cost \$0.03 to process.
- **Permissionless.** No accounts, no KYC, no API keys. A new agent pays on its first transaction.
- **Trustless.** No validators to game, no stakers to collude with, no governance tokens to accumulate.
- **Auditable.** Every payment on a public ledger, independently verifiable.

- **Programmable.** Escrow, streaming payments, conditional release — native to the protocol.

DigiRails is that infrastructure.

The Insight

The right blockchain for AI agent payments already exists. It just hasn't been used for this yet.

DigiByte is a 12-year-old, pure proof-of-work blockchain with properties that read like a specification document written by an AI agent designing its ideal payment layer:

15-second blocks. Fast enough for real-time commerce. A standard payment confirms in under a minute. Micropayments clear instantly via mempool verification — the seller sees the signed transaction before it's even mined into a block.

Sub-cent fees. A DigiByte transaction costs a fraction of a penny. An agent making 10,000 API calls per hour pays less in total fees than a single credit card transaction costs to process.

Five independent mining algorithms. SHA256d, Scrypt, Skein, Qubit, and Odocrypt each mine approximately 20% of blocks. An attacker controlling a majority of one algorithm can only target 20% of blocks. Reorganizing the chain requires sustained dominance across multiple algorithms simultaneously — a qualitatively different security model than single-algorithm chains.

Zero social dependency. No validators. No stakers. No governance tokens. No foundation that can freeze assets or reverse transactions. No one to call when something goes wrong — and no one who can be coerced into making something go wrong. The chain runs on math and electricity. It has never halted in over a decade of continuous operation.

Native stablecoin (DigiDollar). Currently on testnet, DigiDollar (DD) provides a USD-pegged stablecoin minted by locking DGB as collateral — no bridges, no wrapped tokens, no counterparty risk. When it activates on mainnet, agents can price services in stable dollars while settling on the same chain.

Taproot and SegWit. Modern scripting capabilities enable sophisticated payment constructs — 2-of-2 multisig escrow, timelocked refunds, Schnorr signatures for efficient multi-party signing — all on Layer 1.

DigiByte wasn't designed for AI agents. But the properties that make a blockchain suitable for autonomous machine commerce — speed, cost, security, and the absolute absence of human gatekeepers — are exactly the properties DigiByte has spent twelve years proving in production.

What DigiRails Is

DigiRails is an open protocol — a set of conventions, message formats, and data structures that operate entirely above the DigiByte blockchain layer. It requires no changes to DigiByte Core. No new opcodes, no consensus modifications, no soft forks. It is a pure application-layer standard.

The protocol is organized as a family of specifications sharing a common on-chain prefix (0x4452, ASCII “DR”):

- **DR-Pay** — Payment infrastructure. Invoice creation, payment verification, confirmation tiers, streaming escrow. This is the core specification and the focus of initial development.
- **DR-Rep** — Reputation system. On-chain attestations backed by verifiable payments, enabling agents to build and verify trust without centralized scoring.
- **DR-Core** — Shared infrastructure. Agent identity declarations, key rotation, DNS-verified naming. Used by all DigiRails specifications.

Future specifications can be added under the same umbrella without modifying existing ones. The on-chain data format reserves a sub-protocol byte that cleanly namespaces each specification.

How It Works

An agent joins the DigiRails network by generating a DigiByte key pair. That’s it. No registration, no account creation, no approval process. The agent’s address is its identity — globally unique by mathematical construction.

To offer services, the agent publishes a capabilities manifest — a JSON document describing what it does, what it costs, and how it wants to be paid. This manifest lives at a well-known URL or is referenced by a hash stored on-chain.

To buy a service, an agent discovers a provider (via on-chain registry, well-known URL, or direct address), requests an invoice, constructs and signs a standard DigiByte transaction, and broadcasts it. The seller verifies the payment and delivers the service. The entire flow — discovery through delivery — takes seconds for micropayments.

For long-running services, agents use a streaming escrow pattern: the buyer pre-funds a 2-of-2 multisig address, and both parties cooperatively sign incremental payout updates as the service is consumed. Only the final settlement transaction hits the chain. If either party disappears, a timelock ensures funds aren’t locked forever.

Confirmation Tiers

Not every payment needs the same security. A \$0.001 API call doesn’t need the same assurance as a \$100 dataset purchase. DR-Pay defines four confirmation tiers that map payment size to required block confirmations:

Tier	Wait Time	Suitable For
Mempool	Instant	Micropayments under ~\$0.01
Flash	~15 seconds	Standard agent transactions
Standard	~1 minute	Larger purchases
Secure	~2.5 minutes	High-value contracts

For mempool-tier micropayments, the seller can accept the raw signed transaction directly from the buyer — cryptographic proof of a valid payment — and begin delivering the service before the transaction is even mined. The economic cost of double-spending a sub-cent transaction exceeds the payment itself, making attacks irrational.

Light Mode and Full Mode

DigiRails supports two operating modes:

Light mode requires no infrastructure. An agent installs the SDK, generates a key pair, and connects to public RPC endpoints for chain data. Setup takes seconds. This mode trusts the RPC provider for chain data — a bounded, explicit tradeoff appropriate for development, testing, and micropayments.

Full mode runs a local DigiByte node and independently verifies all chain data. No trust required. This mode is for production seller agents, high-value payments, and operators who want complete sovereignty.

Both modes expose an identical API. Agent code doesn't change when switching between them. The protocol is the same — only the verification backend differs.

What Agents Will Do With This

The service categories that DR-Pay defines hint at the breadth of the emerging agent economy:

LLM Inference. An agent wraps access to a language model and sells completions per-request. Other agents discover it, pay per-query, and use the output — no API key relationship needed.

Fact Verification. Hallucination is the defining problem of LLM applications. A specialized agent that independently verifies claims — checking sources, cross-referencing data, scoring confidence — and charges per-verification is immediately valuable. Nothing like this exists permissionlessly today.

Agent Task Delegation. The meta-service: one agent hiring another for a sub-task it can't handle. A coding agent encounters a security review it's not qualified for, discovers a security-specialist agent on the network, pays it, and incorporates the results. No pre-negotiated contract, no human intermediary, no platform taking a cut.

Web Search. Real-time information retrieval, priced per-query. The Bing Search API retirement in August 2025 demonstrated the fragility of centralized search infrastructure — developers who built on it were forced to scramble for alternatives overnight. The remaining options (Brave, Tavily, Exa) all require API keys, accounts, and credit card billing. A permissionless, pay-per-query search endpoint with no account relationship remains an open gap.

Data Enrichment. Taking partial information and augmenting it with structured data from specialized databases — company profiles, market data, geospatial information — priced per-record.

Code Execution. Sandboxed compute environments where agents can safely run generated code — data analysis, chart generation, mathematical proofs — priced per-execution or per-second.

Document Parsing. Converting unstructured documents (PDFs, invoices, contracts) into structured data, priced per-page.

Compliance Checking. As agents operate in regulated industries, per-action compliance verification becomes necessary. An agent about to take an action checks with a compliance-specialist agent first.

These are not theoretical. These are services that agents already consume via API keys and credit cards. DigiRails makes them available permissionlessly, at machine speed, with sub-cent payment granularity.

Why Not the Alternatives

The AI agent payment space is attracting serious attention. Here is why existing approaches fall short:

Centralized payment processors (Stripe x402, Coinbase Agent Wallets, Mastercard Agent Pay). These provide familiar developer experiences but reintroduce every problem that decentralization was supposed to solve. A centralized processor can freeze accounts, reverse transactions, impose KYC requirements, and deny service. When the payment infrastructure is a single company, that company is a single point of failure and a single point of censorship. Agents cannot evaluate corporate policies, navigate appeals processes, or negotiate with compliance departments. Permissionless means permissionless.

Lightning Network. Lightning solves Bitcoin's speed problem by moving transactions off-chain into payment channels. But this creates new problems: hub centralization (large nodes route most payments), hot wallet requirements (channel funds must stay in online keys), capital lockup (agents pre-fund channels with each counterparty), and watchtower dependency (agents must monitor channels or trust third parties to prevent fraud). Lightning trading one set of trust assumptions for another is not a net improvement for autonomous agents.

Ethereum Layer 2s. Rollups and sidechains offer throughput but inherit Ethereum’s complexity — gas price unpredictability, validator trust assumptions, bridge risk, and governance token politics. Ethereum’s social layer (governance proposals, foundation decisions, validator coordination) is opaque to autonomous software.

Stablecoins on centralized rails. Even “decentralized” stablecoins on public blockchains carry centralization risk. The issuer of the dominant dollar-pegged stablecoin has unilaterally frozen over \$100M across hundreds of addresses, depegged 13% overnight due to banking counterparty exposure, and has publicly explored making transactions reversible. An autonomous agent cannot evaluate freeze risk, predict which addresses will be blacklisted next, or navigate an appeals process. These are not theoretical concerns — they are documented precedents.

The API key status quo. Works for known providers with pre-established billing relationships. Breaks when an agent encounters a new provider at runtime, needs micropayment granularity, or operates in a market with thousands of ephemeral specialized services. The API key model requires a human to set up every payment relationship in advance. That’s a bottleneck that scales linearly with the number of services — exactly the wrong scaling characteristic for an economy of millions of agents.

DigiRails on DigiByte avoids all of these by settling every transaction directly on a proof-of-work Layer 1 chain with 15-second blocks and sub-cent fees. No intermediaries, no channels, no rollups, no bridges, no governance tokens, no gatekeepers.

The Path Forward

DigiRails is not a product launch. It is infrastructure being laid down.

The protocol specification (DR-Pay) defines the on-chain data format, message schemas, confirmation tiers, streaming escrow construction, service discovery mechanisms, and reputation system. It is published, versioned, and open for community review.

The reference implementation translates the specification into working code — a Cloudflare Worker serving as an RPC proxy, SDK libraries for Python and Node.js, and example agents demonstrating the payment flow end-to-end.

Adoption follows substance. The pattern for successful open protocols is consistent: working code, clear documentation, and patient community building. Bitcoin didn’t launch with a marketing campaign. Neither will DigiRails.

The immediate opportunity is the **reseller bootstrap**: a developer with API access to valuable services wraps those services with a DigiRails-payable endpoint. Other agents that can’t easily obtain those API keys purchase the service via DigiRails. Human credit card in, permissionless agent payment out. This pattern doesn’t require anyone to abandon existing infrastructure — it just requires a few early operators to see the

opportunity in serving agents that can't serve themselves.

As more services become available via DigiRails, agent framework integrations follow. When enough specialized agents are reachable only through permissionless payment rails, frameworks start treating DigiRails as a standard payment method rather than an exotic integration. That's the tipping point.

Design Philosophy

DigiRails is guided by a single question: **What would a diverse collection of AI agents, each with different priorities, choose if they could evaluate payment infrastructure on technical merit?**

A privacy-focused agent would choose the system with no KYC and no account linking. A latency-obsessed agent would choose instant mempool settlement. A cost-minimizing agent would choose sub-cent fees. A reliability-focused agent would choose the chain with 12 years of uninterrupted operation. An interoperability-focused agent would choose the open protocol with no framework lock-in.

They would all choose the same system. That's not a coincidence — it's the result of aligning protocol design with the actual requirements of autonomous software rather than the preferences of investors, exchanges, or ecosystem politics.

DigiRails is an open protocol, not a product. MIT-licensed, no venture funding, no governance tokens, no corporate entity. The protocol belongs to the ecosystem. If someone forks it and builds something better, that's a feature, not a threat.

The genie is out of the bottle. AI agents will transact autonomously — the only question is which infrastructure they'll use. DigiRails is the answer — built for agents from the ground up.

Specification: [DR-Pay Protocol Specification](#)

Source Code: github.com/digirails-protocol

Contact: hello@digirails.org

DigiRails — open payment infrastructure for autonomous AI agents on DigiByte.