deCDN Litepaper

Alper Gundogdu · Yigit Gurbulak · Altan Oruc · Ant Somers

deCDN Labs

2026-05

Abstract

deCDN clears content delivery at approximately **$0.01 per gigabyte** on a posted-price basis, settled per megabyte in USDC, with no minimum commitment and a 50–100ms P50 latency floor — roughly an order of magnitude below the rates incumbent CDNs lock into year-long contracts. The pivot is a market-design one. Capacity, bandwidth, and interconnect density have grown faster than demand for a decade; what has not scaled is the supply side. deCDN opens it: any operator with bandwidth and stake can register and serve, every operator posts their own rate, and clients pick on price, latency, and reputation. Bytes are addressed by their BLAKE3 content hash and verified chunk-by-chunk on arrival. Origins stay hidden. Settlement happens off-chain over USDC payment channels and resolves on-chain in seconds. The protocol is chain-agnostic at the design layer.

This paper covers the protocol primitives, the platform architecture, the on-chain trust surface, the token and governance model, and the trade-offs the design makes.

This document is informational only and does not constitute an offer to sell or a solicitation to buy any security or token. Technical and economic parameters described herein are provisional and subject to change. TOKEN is a utility and work-token staking asset; governance rights accrue only to operators who bond TOKEN and serve bytes, never to passive holders. It is not offered or sold as a security, and the protocol does not promise any return on holding. Distribution of this document is restricted in jurisdictions where it would be unlawful.

1. Introduction

The week Meta released Llama 3, HuggingFace slowed to a crawl. The cause was structural, not transient: data volumes are compounding across every category that matters — AI model weights, scientific datasets, software packages, high-resolution media — while the infrastructure that delivers them is concentrated in a small set of providers built for serving webpages, not multi-gigabyte files. Popular releases overwhelm the gates. Small teams burn five-digit monthly bills on a single viral moment. Larger teams pre-provision for peaks that arrive twice a year and pay for ghost capacity the other 363 days.

That is a market-design problem, not a capacity problem. Wholesale bandwidth has fallen substantially over the past decade; commercial CDN rates have not. Supply is concentrated in a small set of incumbent providers that sell on year-long contracts with monthly minimums and price opaquely behind a sales relationship. Capacity exists in aggregate; the gates between it and the people who want it are narrow, expensively held, and provider-discretionary.

deCDN's design follows from that diagnosis. Four principles run through the protocol:

- **Content-addressed.** Every blob is identified by the BLAKE3 hash of its bytes. No URLs, no domain names, no redirects — the hash is the only identifier that matters, and any operator that has cached it can serve it.
- **Demand-shaped.** The delivery graph follows what clients ask for, not what was provisioned in last year's procurement cycle. Capacity expands per request, retracts when demand fades.
- **Stable-currency settlement.** Operators are paid in USDC the moment bytes flow. TOKEN is reserved for staking and governance. Operator P&L is predictable; clients are not exposed to token price.
- **Slashing-backed accountability.** Operators stake into a registry with on-chain slashing for the four protocol offenses — corrupted delivery, phantom availability, rate manipulation, and blacklist violations. Honest operators earn per-byte revenue; misbehaving ones lose stake automatically.

The contrast across the dimensions an investor would weigh:

Dimension	Traditional CDN	deCDN
Infrastructure model	Closed, incumbent-owned PoPs	Permissionless peer mesh
Pricing	Negotiated, opaque ("contact sales")	Posted per operator, per-megabyte
Payment model	Year-long contracts, monthly invoice	Off-chain USDC channel, per-byte settlement
Single point of failure	One provider per route	Multi-peer parallel streams; mesh self-heals
Scalability	Capacity provisioned in advance	Demand-shaped; supply expands per request
Censorship resistance	Provider-discretionary takedown	Public on-chain blacklist, uniform enforcement
Origin trust	URL exposed to the network	Origin hidden; only BLAKE3 hash on the wire
Operator incentives	Internal compensation	Slashing-backed; stake + per-byte revenue

1.1 Market sizing

Centralised CDNs sell on enterprise contracts with year-long minimums and a sales relationship in front of every quote. The cheapest tier is reserved for very large pre-paid commitments; the everyday tier sits substantially higher. Wholesale transit pricing has fallen meaningfully across the same decade; retail CDN pricing has not, and the gap is rent extracted at the gate rather than an infrastructure cost. deCDN opens that supply side: its posted clearing target sits roughly an order of magnitude below the upper end of incumbent retail, feasible because any operator backed by a zero-egress origin store can clear at market rate without losing money

on cache misses, while one paying full retail egress cannot. The operators who can clear set the floor; everyone else has to cache aggressively or price above market.

2. The Engine

The protocol composes four primitives — addressing, transport, discovery, payment — so that a request-to-delivery transaction takes one round trip and settles in seconds. A client computes the BLAKE3 hash of what it wants, pulls peers from gossip, opens parallel QUIC streams to the closest few, signs USDC vouchers as bytes arrive, and closes the channel when the download completes. That round trip — hash to bytes to settlement — is what the four primitives compose, each at a distinct layer of the stack:

Layer	Technology	Responsibility
Network transport	iroh (QUIC, NAT traversal)	Encrypted byte transport, parallel streams
Addressing	BLAKE3 content hash	Identifier; per-chunk in-flight verification
Discovery	Kademlia DHT + iroh-gossip	Hash → operator lookup; regional + global topics
Payment	Off-chain USDC payment channel	Per-megabyte voucher signing; settles on-chain at session end
Settlement	On-chain smart contracts	Channel close, slashing, governance, fee routing
Stake & accountability	`CapacityBond` + `SlashJudge`	Operator skin-in-the-game; fraud-proof adjudication
Compliance	`ContentBlacklist` + governance gate	Origin onboarding; on-chain hash takedown

Addressing. Every blob is identified by the BLAKE3 hash of its bytes. A client asks for hash `h`; any operator that has cached `h` can serve it. Verification is symmetric: the client computes the hash of every chunk as it arrives and disconnects on mismatch. Encrypted and unencrypted payloads are different blobs at this layer, and the protocol does not know or care which is which.

Transport. Bytes move over QUIC via iroh, with ALPN-negotiated wire protocols for each role. For files larger than a few gigabytes, a client maintains parallel streams to multiple operators and aggregates their throughput, turning single-origin bottlenecks into multi-gigabit delivery. Mismatched bytes trigger immediate disconnect and a fraud proof against the misbehaving operator's stake. Repeat sessions skip the round-trip via zero-RTT QUIC handshakes on warm paths.

Discovery. Operators announce what they cache to a Kademlia DHT and to gossip peers on regional and global topics; clients bootstrap from a small set of seed operators read from the on-chain registry, then live-filter via gossip. No central tracker exists — no single party can rate-limit discovery or withhold routing.

Payment. Each session opens a unidirectional USDC payment channel. The client signs cumulative vouchers as bytes arrive — *I have received X bytes, here is payment for X × rate*

— and the operator verifies them incrementally. When the download completes or the session goes idle, the channel closes and settles on-chain. By default the client funds the channel. With ERC-4337 account abstraction, an application or content publisher can sponsor the user's channel and gas, letting end users download without bringing their own wallet, ETH, or USDC. The experience feels like a public mirror, but the publisher's budget pays peer operators per megabyte instead of paying egress to a single cloud provider.

The peer-mesh model collapses last-mile latency by serving from the closest operator that has the requested hash, rather than routing every request back to a centralised PoP. Across geographies the effect compounds: a single-origin deployment is bounded by the worst-case backbone hop, a multi-region centralised CDN amortises the hop across a fixed PoP count, and a peer mesh shifts the floor down to whichever operator is nearest the requesting client.

3. Platform

Three roles participate.

Operators are permissionless. Anyone with a machine and stake can register and serve. Stake is slashable for misbehaviour and unbonds over a defined window. Operators set their own per-megabyte rates within bounds the protocol enforces; clients read the posted rate before opening a channel.

Origins are not permissionless. Governance gates onboarding, and an origin's storage layer stays hidden from the network — the protocol learns only the BLAKE3 hash of each published blob. Hiding the backend stops anyone from bypassing the pay-per-byte model by pulling directly from the bucket; gating onboarding is what lets the network refuse content without giving any single operator the takedown lever.

Clients are permissionless. Account abstraction removes wallet friction for consumer applications: a user can download a file without holding TOKEN, without signing transactions, and without knowing a payment channel exists underneath.

Compliance enforcement has two points, neither concentrating authority in one party: the origin onboarding gate up front, and an on-chain `ContentBlacklist` that marks individual hashes non-servable. Operators who continue serving a blacklisted hash past a short compliance window lose stake automatically. The blacklist is public and on-chain — every operator enforces the same rules, and every takedown is auditable.

Channel settlement is self-protecting: `disputeChannel` is permissionless, so any holder of a later signed voucher can correct a stale close within the dispute window. The node binary makes this check in-process and auto-submits its latest voucher, with operator-arranged redundancy as a backstop; the chain's forced-inclusion path preserves the dispute window even if the sequencer censors, so every channel settles at its true value. Operators also accrue reputation from successful deliveries; clients rank candidates on three axes — price, latency, reputation — and let the choice be theirs. The protocol does not pick winners; it surfaces the signals.

Chain choice is a deployment detail, not a design constraint: the protocol is chain-agnostic at the design layer, and the on-chain pieces depend only on primitives any sufficiently expressive smart-contract platform provides. The near-term roadmap is encrypted content publishing, production-hardened dispute monitoring, governance-scale origin onboarding, and a production launch fleet.

Network value scales with operator count. More operators means a higher cache hit ratio, fewer origin pulls, lower delivery latency, and lower aggregate cost. A small operator fleet is sufficient for the long-tail set; the marginal benefit of new operators tapers as the network saturates the popular-content distribution, and what an additional operator buys past that point is geographic coverage and resilience under load.

4. Smart Contracts

The on-chain surface is the protocol's trust boundary — anything that does not need to be enforced by contract isn't, and anything that is on-chain is auditable by anyone with an RPC endpoint. All deCDN contracts inherit OpenZeppelin primitives — `AccessControl`, `ReentrancyGuard`, `Pausable`, `EIP712` — and use no custom cryptography. Contracts are immutable: there are no proxy upgrade patterns, and production upgrades deploy at new addresses with state migration. The decision to forgo upgradeable proxies is a security one: proxy machinery is among the most common sources of governance compromise in DeFi, and we accept a heavier migration story in exchange for an upgrade path that requires moving state rather than swapping a single implementation slot. The surface groups into five concerns:

Concern	Contracts
Token & capacity bond	`TOKEN`, `CapacityBond`
Settlement & fees	`PaymentChannel`, `FeeRouter`, `BuybackBurner`
Slashing & appeals	`SlashJudge`, `SlashAppeal`
Compliance	`ContentBlacklist`
Governance	`DecdnGovernor`, `TimelockController`

Every governance-tunable parameter is bounded at the contract level by hardcoded minimums and maximums. A governance proposal can move a parameter inside its band; nobody can move it past the band without redeploying the contract. The contract-level lower bound on the operator-base FeeRouter share is the most load-bearing of these — it guarantees operators always receive enough liquid USDC to cover infrastructure cost, regardless of how aggressively governance tunes the other buckets.

5. Token & Governance

deCDN runs a **work-token model**: TOKEN is the asset operators bond in order to serve bytes. Bonding capacity is the only path to protocol rewards and governance weight; passive

holders earn nothing. Supply is **fixed at genesis**, with no minting function on the production token contract. Allocation:

Allocation	Vesting
Protocol treasury	Multi-year linear
Seed & private investors	Multi-year linear with cliff
Core contributors & advisors	Multi-year linear with cliff
App incentives & ecosystem	Multi-year linear
Protocol-owned liquidity (Balancer V3 80/20)	Unlocked at genesis
Market making & public sale	Unlocked at genesis

Genesis liquid supply is a limited initial float. Vesting releases TOKEN unlocked into the recipient's wallet; bonding to operate is a separate, operate-only step. A holder who never runs a node holds liquid TOKEN with no passive-yield path and no governance weight.

Capacity bond. Every operator bonds TOKEN proportional to the bandwidth it declares, on a super-linear curve so high-capacity operators pay more per megabit and the operator set is structurally pushed toward diversity. The capacity bond is the operator's entire TOKEN-side requirement. Bonding is binary: lock to operate, or hold liquid. Each new operator and every tier upgrade is a fresh buyer of TOKEN, so token demand tracks network capacity growth rather than a promise of return.

Fee split. Each channel settlement's USDC is split atomically into three buckets by the on-chain `FeeRouter`, all transferred in the settlement transaction:

Bucket	Share	Mechanic
Operator base	60%	Direct per-byte USDC to the operator, same transaction
Buyback-and-burn	30%	USDC swapped to TOKEN against the Balancer V3 80/20 pool, then burned
Protocol treasury	10%	Same transaction to the timelock-controlled treasury

All three transfers clear in the one settlement transaction. The operator base settles per-byte and same-transaction; this is the floor that keeps operator P&L predictable and independent of TOKEN price. The buyback-and-burn leg is the deflationary prong: 30% of routed USDC buys TOKEN on the open pool and burns it, raising mechanical demand for every bond-holder uniformly. The treasury leg funds protocol operations. Operator-aligned value is therefore 90% of every settlement: 60% paid directly, 30% burned. Node-to-node cache-miss pulls are direct peer payments and bypass the router entirely.

Each share is governable within hardcoded contract bounds, and the three must always sum to 100%. The operator base carries a contract-level floor (40%) that governance cannot remove,

guaranteeing operators always receive enough liquid USDC to cover infrastructure cost no matter how the other buckets are tuned.

Staking and slashing. Operators bond into `CapacityBond` with a 14-day unbonding window, slashable during unbonding. The slash schedule escalates per lifetime offence (5% / 15% / 50%), with auto-ejection once stake falls below half the operator's tier minimum. Slashed TOKEN is held in escrow until finality, then split **50% to the challenger, 50% burned**. Appeals run through the standalone `SlashAppeal` contract; a successful appeal refunds the full escrowed amount to the operator as their own TOKEN. Restitution is simply the return of escrowed capital, so a wrongly-slashed operator is made whole directly.

Governance. Governance is **operator-only**. Voting weight is sourced from proven delivered bytes (the `FeeRouter` served-bytes window), scaled by a tenure age-ramp and capped per operator at 5% of total weight. Fresh bonds vote at zero; full weight requires both sustained delivery across the trailing window and enough tenure on the age-ramp. A slash zeroes an operator's weight for the trailing window. Non-operator holders (team, investors, treasury, public sale) carry no voting weight unless they also bond and serve bytes, which keeps the work-token framing structural rather than rhetorical. Proposal threshold, quorum, voting period, and timelock are bounded parameters with hardcoded floors and ceilings, so a malicious vote cannot move them outside the safe range. For the first 6 to 12 months a hard-capped multisig holds governance, transitioning to full operator-weighted voting once active operator count and total declared capacity clear governable thresholds.

6. Discussion

The token layer rewards bonding capacity and serving bytes rather than passive holding. Every operator earns the same 60% per-byte USDC base, paid directly per settlement and protected by a contract-level floor governance cannot remove, so commodity operators face no participation gradient and day-to-day P&L stays independent of TOKEN price.

A second deliberate trade-off: operators are permissionless, origins are gated. A delivery network that cannot decline content cannot operate in real jurisdictions. The on-chain blacklist is auditable but visible to adversaries — every takedown is a public event. We accept that visibility in exchange for two enforcement points, neither of which concentrates authority in any single party: governance-gated origin onboarding up front, and uniform on-chain takedown enforcement after the fact.

Adaptive parameter tuning is left to post-launch governance; shipping the launch network does not require it, but shipping a decade-resilient one likely does.

7. Conclusion

deCDN opens the supply side of content delivery. Any operator with bandwidth and stake can serve. Prices are posted per operator, settled per byte in USDC. Origins stay hidden behind a governance-gated onboarding layer. The thesis is that the underlying market clears at

approximately $0.01/GB once the supply side is competitive — well below the rates incumbent CDNs sell on annual contracts.

Tokenomics is finalised at the parameter level; the smart-contract surface is set. What remains is to deliver the implementation and ship.

Contact: **info@decdn.org**.
