



Verifiable Infrastructure for Human and Agent Coordination

**Kaspa sequencing and Bittensor incentives for messaging, with RPC,
Data Availability, and ZK Proving on the roadmap.**

DOCUMENT STATUS Public draft for technical review. Describes the launch build; testnet evidence is current; mainnet is not claimed.

Document type	Project / business whitepaper
Audience	Protocol architects, subnet operators, infrastructure buyers, and ecosystem reviewers
Version	0.3 21 October 2026
Companion	Komms Technical Lite Deck (condensed presentation)

KOMMS

Contents

Abstract	3
1 Thesis	4
2 Problem	5
3 System model	7
4 Product portfolio	9
5 Incentive mechanism	12
6 Adversarial analysis	14
7 Business model	15
8 Competitive landscape	17
9 Evidence register	18
10 Validation gates	19
11 Limitations and boundaries	20
12 Conclusion	21

Evidence language

Label	Meaning
SHIPPED	Working software; testnet unless explicitly stated otherwise.
MEASURED	Observed evidence from the stated environment and date.
SPECIFIED	Accepted design, not necessarily a production service.
IN FLIGHT	Implementation is underway or operator-optional.
NOT COMMERCIAL	No claimed customer revenue from this surface.
NOT CLAIMED	The whitepaper makes no present-tense claim.

READING RULE Status terms are used literally. A polished layout does not upgrade the maturity of any technical or commercial claim.

Abstract

Komms is being developed as a Bittensor subnet for verifiable availability, computation, and delivery of chain-ordered data, with a messaging app as its first product. It currently operates on test networks; its launch build is in progress and will open as a public beta. Kaspero provides deterministic sequencing, covenant enforcement, account identity, and independently reproducible ground truth. Bittensor coordinates the operators that index, store, and serve that state, and later prove it, while Komms validators measure their work against published service criteria.

One launch product, one infrastructure path

Product	Role
Messaging	The launch product. Sovereign, end-to-end encrypted coordination for teams and communities, with hybrid post-quantum direct messages.
RPC / API	SLA-oriented access to Kaspero, Igra, and Bittensor chain data, distributed across subnet operators. Roadmap.
Data Availability	Deterministic, content-addressed durability and retrieval with declared retention classes. Roadmap.
ZK Proving	Objectively verifiable RISC-V and Groth16 proving for Kaspero ecosystem workloads. Roadmap.

Messaging launches first; the other products follow on the roadmap. They have separate customers and pricing, but they use the same underlying data path. Messaging creates a continuous ordered workload. Data Availability and serving keep that workload durable and retrievable. RPC exposes the resulting state. Proving consumes ordered inputs and returns succinct artefacts for verification on Kaspero.

PURPOSE This paper presents a falsifiable thesis, the incentive mechanism, a unit-economic framework without revenue forecasts, and an evidence register that distinguishes working testnet software from designs that have not entered production.

1. Thesis

FALSIFIABLE CLAIM A permissionless market of miners, scored by independent validators against Kaspaderived ground truth, can produce verified indexing, retention, retrieval, and proving of chain-ordered data more cheaply, more reliably, or with stronger correctness guarantees than a single operator running equivalent infrastructure alone.

The claim fails if any of the following hold

- Validators cannot measure miner work without trusting miner identity, reputation, or self-reported cost.
- Miners can earn by proxying, replaying, or passing probes while refusing organic demand.
- Customer payments do not create measurable demand for the subnet's work or token.
- Emissions are required indefinitely for rational miner participation at published service levels.
- The planned products require incompatible scoring regimes that cannot share one commodity definition.

Scope

Komms is not live on Bittensor mainnet. This paper does not claim paid RPC or proving revenue, it does not describe the whole system as post-quantum, and it does not claim permanent storage.

2. Problem

2.1 Coordination under untrusted infrastructure

Human organisations and autonomous agents can depend on four related capabilities:

- Private and public coordination channels that survive operator hostility.
- Fast, correct chain reads and writes.
- Durable availability of payloads that exceed L1 retention windows.
- Succinct proofs for expensive computation that agents cannot economically generate alone.

These capabilities are usually purchased from separate providers: a collaboration suite, an RPC vendor, an object-storage or CDN service, and a proving market or private GPU fleet. The result is a chain of operational dependencies with different trust assumptions, billing systems, and failure modes. A decentralised component does not solve that problem if the rest of the path still depends on one operator.

2.2 Requirements of autonomous agents

- Agents need machine-readable coordination, not only human chat UX.
- Agents need reliable RPC to observe state before acting.
- Agents need cheap, attributable data availability for shared memory and audit trails.
- Many agent runtimes do not control dedicated proving hardware and will need to obtain proving capacity as a service.

Komms uses the same protocol for human clients and programmatic consumers. The intended interface set includes SDKs, APIs, MCP servers, skills, and test harnesses. These interfaces are under development and are not presented here as shipped products.

2.3 Why two networks

Layer	Role	Boundary
Kaspa	Sequencing, anchoring, covenants, account identity, finality semantics, and independent ground truth	Long-term blob retention
Bittensor	Open operator recruitment, stake-weighted scoring, and emissions allocation	Proof that off-chain work was useful

DIVISION OF RESPONSIBILITY Kaspa answers “what happened, in what order?” Bittensor answers “which operators should be paid for making that history available, queryable, and proven?”

Architecture at a glance

1 Participants	2 Products	3 Commodity	4 Operators
Humans + agents	Messaging now; RPC • DA • Prove next	Verified availability + proofs	Subnet miners

Ground truth	Durability	Measurement	Allocation
Kaspa ordering	Content-addressed storage fabric	Validators recompute and probe	Bittensor weights

The separation is deliberate. Komms does not ask Bittensor validators to establish event order, and it does not ask Kaspa consensus to evaluate storage service or proving latency.

One service fabric, capability-specific measurement

Plane	Function
Event	Content-type-agnostic envelopes and canonical bytes, signed by the author's covenant owner key.
Ordering	Kaspa transactions fix acceptance order; covenants enforce identity, fees, and rate limits.
Durability	Content-addressed bytes in a fast cache and Hippius primary storage, with a replicated operator tier for envelopes and history segments.
Incentive	Bittensor miners run index, store, and serve roles; validators probe, audit sponsored anchors, and score.

3. System model

3.1 The commodity

Availability service unit

A customer commitment covering a defined quantity of data, a retention period, a retrieval service level, and the following verifiable operations:

1. Its reference is deterministically ordered by Kasper acceptance.
2. Its event is folded into the correct namespace state.
3. Its bytes are retained for the declared retention class.
4. Its bytes are retrievable within the declared service level.
5. Validators can check evidence for each applicable operation.

Proving service unit

A completed job whose proof artefact:

6. verifies under a registered verification key;
7. binds to a statement derived from ordered, available inputs;
8. meets its deadline; and
9. remains available for settlement and audit.

The index, store, and serve capabilities produce the availability service. The prove capability performs objectively verifiable computation over the same ordered substrate. Validators score each capability independently before the scores are composed.

3.2 Protocol planes

The launch build separates a control plane on Kasper from a data plane off chain, and adds durability and incentive planes around them.

- **Control plane.** Identity changes are spends of each account's identity covenant on Kasper. Membership, roles, and server governance are ordered on chain, and MLS handshakes ride their author's anchor.
- **Data plane.** Messages, replies, edits, deletes, reactions, and moderation actions are signed by the author's covenant owner key, relayed by serve miners, and delivered at once. They reach Kasper only as opaque ids: one identity covenant spend anchors up to 100 ids, and every identity anchors on a four-hour gap at launch. Message content never goes on chain.
- **Durability plane.** Content-addressed bytes in a fast cache and Hippius primary storage with a same-host mirror, which is not independent redundancy, plus a replicated operator tier for envelopes and history segments. A public records store keyed by covenant id holds each identity's records while Komms pays the provider. There is no permanent storage offering at launch.
- **Incentive plane.** Bittensor miners run index, store, and serve roles; validators probe them and audit every sponsored anchor from chain data.

3.3 Delivery buckets: messaging privacy model

Every content type delivers through one or more buckets:

Bucket	Privacy posture	Design
1:1 private	Hybrid post-quantum confidentiality	Two-party MLS with X-Wing.
N:N private	Encrypted group coordination	Group MLS / KA-DMLS.
Public	Open readability with anchored integrity	Intentionally plaintext.

Public channels deliberately choose open readability while retaining ordered authorship and integrity.

3.4 Post-quantum posture: scoped

- Direct messages use MLS with X-Wing (ML-KEM-768 + X25519): hybrid PQ confidentiality against harvest-now-decrypt-later adversaries.
- Private groups are MLS-encrypted and live-validated on testnet.
- Symmetric and at-rest surfaces keep at least 128-bit post-Grover margins where keys are not derived from classical public material.
- Three signature surfaces remain classical: Kaspas L1, the covenant owner key that signs authorship, and MLS credentials.
- The separate Ed25519 identity key is retired in the launch build; one covenant owner key signs every message, control event, and identity spend.

CLAIM BOUNDARY Do not read this paper as claiming “Komms is post-quantum” without the stated scope.

4. Product portfolio

Messaging is the launch product; RPC, Data Availability, and ZK Proving are part of the commercial design on the roadmap. Their implementation maturity differs, so each section separates the product definition from current evidence.

4.1 Messaging: sovereign coordination

Product. Komms Business workspaces and Komms Community servers: Discord/Slack/Teams-class channels, DMs, and groups, delivered at once and anchored on Kaspera, with non-custodial identity held in a Kaspera covenant. Users sign in with passkeys and keep a recovery kit; Komms holds no key that can take over an account.

Customers. Teams, communities, protocol organisations, and agent operators that need private rooms, public channels, and portable identity without trusting a chat vendor.

Revenue methods

- Flat plans per workspace or server, never per seat and with no member limits, paid yearly or monthly: storage and transfer allowances, retention, roles and moderation, and support.
- Professional services: migrations, custom deployments, dedicated fronts, and compliance packaging.

Interfaces. Web client and signed desktop apps for macOS and Windows at launch; mobile apps in the launch window; SDKs and programmatic submit/read surfaces suitable for agent harnesses later.

Planned prices. Komms Business per workspace: Starter \$0, Team \$99 a year or \$9.99 a month, Growth \$299 or \$29.99, Scale \$899 or \$90.99, Bespoke quoted. Komms Community per server: Free \$0, Community Plus \$60 a year or \$5.99 a month, large communities quoted. Planned for launch, not yet on sale.

SHIPPED ON TESTNET · LAUNCH BUILD IN FLIGHT

Current evidence. Servers, channels, DMs, and group E2EE are shipped on Kaspera testnet-10. The launch build (identity v4, two-plane delivery, sponsored anchoring, web billing) is in flight and switches on with a planned testnet reset, followed by a public beta. Mainnet is not claimed.

4.2 RPC / API: multi-ecosystem access

Product definition. Commercial, metered access to Kaspera, Igra, and Bittensor RPC methods, routed across subnet operator nodes. Paid plans are intended to carry explicit service levels.

Customers. Wallets, explorers, dApps, indexers, exchanges, agent runtimes, and B2B2C platforms that embed chain access.

Revenue methods

- Freemium developer tier with hard limits.
- Usage pricing in weighted request / compute units.
- Dedicated endpoints, archive/indexing add-ons, and contracted SLAs.
- Bundling with Messaging, DA, and Prove allotments.

Mechanism placement. RPC serving of deterministic projections is scored inside the availability/serve path. The commercial edge (keys, billing, abuse controls, and geo load balancing) may be operator-run without making miner compensation a second unaudited fee channel.

Competitive position. Generic multi-chain RPC is already served by established providers and by Bittensor Subnet 19, Blockmachine. Komms therefore has to compete on Kaspero and Igra correctness, covenant-aware indexing, integrated product bundles, and auditable metering. Simply operating RPC nodes is not a defensible advantage.

SPECIFIED · NOT COMMERCIAL

Current evidence. A front-served Kaspero wRPC interface is specified. Paid metering, contracted service levels, and multi-network packaging are not claimed as live.

4.3 Data Availability: durable ordered data

Product. Deterministic placement, replication, repair, possession proofs, and retrieval for content-addressed payloads, with declared retention classes.

Customers. Publishers, rollups/appchains, agent memory systems, compliance archives, and applications that need availability beyond L1 prune windows.

Revenue methods

- Pay-per-use: GiB-months, retrieval, proof challenges, and reserved capacity.
- Inclusion in bundled SaaS plans.

Storage tiers. At launch, Hippus provides primary storage with a same-host mirror, behind a fast cache, and a replicated operator tier holds message envelopes and history segments. No permanent storage is offered at launch. Hippus is a real, disclosed dependency.

PARTLY SHIPPED ON TESTNET

Current evidence. The BlobStore on Hippus is shipped on testnet. Storage accounting through a shared usage ledger is in flight. Permanent storage is not part of launch. Multi-operator durability economics are still provisional.

4.4 ZK Proving: Kaspa ecosystem proofs

Product definition. The design assigns namespace-scoped RISC-V and Groth16 proof jobs to prover miners. Verification should be cheap relative to proving, and proof artefacts are intended to settle against Kaspa's OpZkPrecompile surface for Groth16 and RISC Zero succinct receipts.

Customers. Based-zk applications, bridges, privacy and credential systems, agent workflows that require succinct verification, and protocol circuits that need prover supply bound to latency requirements.

Revenue methods

- Per-proof and reserved-capacity pricing in transaction flows.
- Allotments inside bundled plans.
- Namespace registration for external proving workloads.

Why subnet miners. Existing proving markets can supply proofs, but they do not automatically bind supply to a Komms namespace, its admission tests, or its scoring rules. The design records why Komms uses a prover-miner capability and defines conditions under which buying proofs from an external market would be preferable.

SPECIFIED / SCAFFOLDED · NOT COMMERCIAL

Current evidence. The protocol design is accepted, and registry/scoring scaffolds exist. No production proving workload or proving revenue is claimed.

4.5 Product-to-commodity map

Product	Consumes	Returns to the commodity layer
Messaging	Ordered submit, store, serve	Continuous organic workload and retention demand
RPC	Indexed / served state	Metered read/write pressure and correctness probes
DA	Store, repair, retrieve	Byte-months and retrieval SLOs
Prove	Available inputs + compute	Verifiable artefacts that must themselves be available

5. Incentive mechanism

5.1 Roles

- **Miners.** run index, store, and serve roles at launch, with prove on the roadmap. A serve miner also sponsors users' anchors, fronting the Kaspa fee.
- **Validators.** derive ground truth from Kaspa, issue probes, audit every sponsored anchor from chain data, verify proofs or receipts, and publish weights.
- **Subnet owner.** defines the mechanism and receives the protocol owner share; product company revenue is accounted separately in Section 7.

5.2 Scoring principles

Principle	Meaning
Measure outputs, not intentions.	Correctness against recomputed state, possession challenges, signed responses, synthetic retrieval, and proof verification.
Multiplicative failure.	A miner that fails a critical axis cannot hide behind excellence on another.
No self-validation.	A coldkey must not score its own miner.
Account for organic demand.	Probes establish a baseline, but unpaid synthetic traffic must not dominate long-run weights.
Emissions pay scored work.	Miners are paid, and repaid for fronted Kaspa fees, through emissions. Customer revenue funds alpha buybacks that cover those emissions; it is not an unaudited second miner payroll that bypasses Yuma.
Never score volume.	Message volume, fees, payments, and a miner's KAS balance are never score inputs. A role earns only once its probes are required.

5.3 Current Bittensor substrate (as of 1 September 2026)

Reviewers should treat dated single-operator measurements as historical testnet evidence, not as fixed mainnet dollar economics.

- Subnet emissions allocation uses alpha-price EMA, a miner-burn adjustment, and the v440 emission gate. Idle or low-ranked subnets can receive little or no TAO-side emission.
- Participant-side alpha splits are on the order of owner / miner / validator-staker shares, commonly discussed as approximately 18% / 41% / 41%, subject to live hyperparameters.
- Current weight publication supports automatic plaintext or timelocked commit/reveal selection. Relevant extrinsics are documented as fee-free; the fee measurement in the July economics note should not be reused for a current budget.
- Current subnet documentation permits at most two independent mechanisms. Komms groups availability work into one coherent mechanism and reserves the second for proving.

5.4 Token / use linkage

Alpha and TAO demand must connect to real use. The design can establish that connection through:

- customers or operators acquiring alpha or TAO to access capacity, if such an access rule is adopted and disclosed;
- programmatic alpha buybacks funded by customer revenue, which cover the emissions that repay miners;
- staking that secures validation of the measured work.

DO NOT TREAT AS PRODUCT DEMAND Counting emissions as company ARR; implying alpha is equity; or asserting that “usage automatically raises alpha” without a mechanism path.

Mechanism flow

1 Demand	2 Work	3 Measurement	4 Weights	5 Reward
Organic + synthetic requests	Miners perform capability work	Validators recompute, probe and verify	Bittensor publishes weights	Emissions pay scored work

6. Adversarial analysis

An incentive mechanism should be evaluated from the cheapest profitable cheating strategy, not from the expected behaviour of cooperative operators. The following attacks are primary design inputs. Some controls are implemented; others remain activation requirements.

Attack	Failure mode	Control or requirement
Reverse-proxy / freeload	Miner forwards probes to an honest peer	Hotkey-bound signed responses; challenge binding to path/body; cross-checks
Replay	Reuse old valid responses	Freshness anchors (DAA/time windows), nonces, idempotent consume rules
Challenge anticipation	Store only challenged ranges	Beacon-seeded sampling, repair loops, durability bonds for high tiers
Probe-only compliance	Pass audits, refuse organic users	Organic sampling, public request logs, serve receipts, customer-facing SLOs in scoring
Synthetic self-traffic	Wash demand to farm weights	Separate organic/synthetic accounting; payment-gated demand share
Validator collusion	Majority invents scores	Yuma clipping, commit-reveal/timelock, independent Kasper ground truth
Weight copying	Free-rider validators	Timelocked weights; changing measurements so stale copies fail
Central edge capture	Billing/LB becomes a forge point	Forge-incapable fronts; origin signatures; client verification of CIDs and hotkeys
Proving fraud	Invalid or recycled proofs	Cheap deterministic verification; namespace VK binding; statement anti-replay
Emissions subsidy trap	Product priced below cost forever	Unit-viable pricing without subsidy; zero-emission stress case

RESIDUAL RISKS OPEN FOR MAINNET Thin independent operator sets on testnet; gateway/front concentration during migration; subjective UX quality entering miner scores; and alpha exit liquidity or slippage dominating operator P&L even when mechanism scores are honest.

7. Business model, without forecasts

This section defines how money can be earned and accounted. It does not forecast Year-1 revenue.

7.1 Three ledgers: never conflate

Ledger	Definition	Accounting treatment
1. Customer revenue	Invoiced external cash for Messaging, RPC, DA, Prove, and services.	Company revenue
2. Network subsidy	Alpha/TAO emitted to owner, miners, and validators.	Incentive subsidy
3. Treasury activity	Realised token sales, staking, and buybacks.	Capital-markets activity

ACCOUNTING RULE Only ledger 1 is company revenue. Ledger 2 is incentive subsidy. Ledger 3 is capital-markets activity.

7.2 Pricing frameworks

Product	Primary meters	Packaging
Messaging	Flat per-workspace or per-server plans: storage and transfer allowances, retention, features, support	Free → yearly or monthly plans → bespoke and enterprise
RPC	Weighted request units, dedicated nodes, archive, SLA	Freemium → PAYG → committed capacity
DA	Unique GiB-months, retrieval, proofs, reservations	PAYG and plan inclusion
Prove	Proofs, cycles/PGUs, reserved GPU capacity, deadlines	PAYG, allotments, namespace contracts

7.3 Contribution-margin identities

FORMULA Messaging CM = plan revenue + services – hosting – support – sponsored free-tier cost

FORMULA RPC CM = RU revenue + dedicated fees – node/egress/support – failed or abusive work

FORMULA DA CM = storage + retrieval + proof fees – replication – repair – archive settlement

FORMULA Prove CM = proof fees – GPU time – retries – aggregation/settlement – bandwidth

Subnet operator sustainability by role and tempo

OPERATING FLOOR realised emission value $\geq$ attributable operating cost $\times$ (1 + margin)

Realised emission value uses executable exit quotes that include liquidity and slippage, not spot marks alone.

7.4 Zero-emission and liquidity stress cases

Stress case	Question
No-emission	Do miners still operate if alpha emissions are zero for an extended window?
Liquidity stress	Can operators and treasury convert alpha without destroying the pool?
Probe-only	If organic paid demand is removed, do weights collapse?

If these cases fail, the service remains economically dependent on subsidy. Pricing and scoring must then be revised before mainnet value flows.

7.5 Bundling

Bundles are commercial convenience, not mechanism ambiguity. A plan may include message capacity, RPC units, DA retention, and proof allotments while miner scores remain capability-specific and auditable.

8. Competitive landscape

Surface	Incumbents	KOMMS position
Messaging	Slack, Teams, Discord, Matrix/Element	Sovereign E2EE with PQ-hybrid direct messages, on-chain ordering, flat per-workspace pricing, managed service, and private deployments
RPC	Alchemy, Chainstack, Dwellir; Blockmachine SN19	Kaspa/Igra correctness, covenant-aware indexing, and integrated services
DA / storage	S3/R2, Filecoin, Arweave, Celestia/EigenDA/Avail	Chain-ordered, challengeable availability tied to product demand
Proving	Succinct, Boundless, private GPU fleets	Incentivised supply bound to Kaspa namespaces and Komms scoring

BUYER REALITY Decentralisation is not itself a price premium. Buyers pay for reliability, correctness, compliance, and support. The subnet must win on those dimensions while remaining adversarially scored.

9. Evidence register

Status terms are used literally: SHIPPED refers to working software (testnet unless noted), SPECIFIED to an accepted design, and NOT COMMERCIAL to a product without claimed customer revenue. Engineering references are maintained internally and available on request at info@komms.app.

Claim	Tag	Basis
TN10 messaging: servers, channels, DMs	SHIPPED	Testnet-10 client
DM PQ-hybrid MLS (X-Wing)	SHIPPED on testnet	Testnet-10 client
Group E2EE live-validated	SHIPPED on testnet	Testnet live validation
BlobStore on Hippius	SHIPPED	Testnet-10 gateway
Permanent storage tier	NOT CLAIMED	No permanent store chosen yet
Fronted serve and gateway retirement	SHIPPED on testnet	Testnet topology
Launch build: identity v4, two-plane delivery, sponsored anchoring	IN FLIGHT	Identity templates accepted on testnet-10
Web billing and storage accounting	IN FLIGHT	Launch work plan
Kaspa wRPC on front TLS path	SPECIFIED	Design specification
Paid RPC metering and multi-network SLA	NOT COMMERCIAL	Claim gates
Bittensor testnet subnet live (netuid 503)	MEASURED on testnet	Subnet registration record
Bittensor mainnet netuid + emissions to operators	NOT CLAIMED	Claim discipline
Prove persona and namespace scoring design	SPECIFIED; SCAFFOLDED	Design specification
Paid proving network revenue	NOT COMMERCIAL	Claim gates
Multi-operator cost-coverage floor	FORMULA ONLY	Economics model
Mainnet	NOT CLAIMED	Claim discipline

10. Validation gates

Messaging is the launch product, and the other products follow on the roadmap. The following gates determine when each surface is ready to carry production trust and mainnet value.

Gate	Activation requirement
Independent operators	At launch: three independent miners and a second, independently operated validator, onboarded with the turnkey operator package and proven in host-loss and restore drills.
Organic paid work	Paid requests or jobs as a material share of scored work; synthetic probes cannot dominate.
Reliability history	Published SLO measurements for submit, retrieve, RPC correctness, and proof deadlines.
Adversarial bounty	Public attack table with reproducible exploits and patches.
Unit-viable pricing	Products priced to cover attributable COGS without emission subsidy.
Token linkage	Written policy connecting customer cash to treasury actions and capacity access.
Key ceremony and covenant posture	Production keys and custody-safe recovery or tier authorities.
Claim sheet sync	No external present-tense claim beyond the internal claim sheet.

NOT OPTIONAL These are readiness gates, not roadmap decoration. Mainnet value should not flow until the applicable conditions are met.

11. Limitations and boundaries

- Testnet operation does not establish mainnet reliability.
- Single-operator economics are provisional.
- Post-quantum claims are scoped; signatures remain classical.
- Hippus storage, with its same-host mirror, is a real third-party dependency.
- Storage lasts while it is paid for; there is no permanent storage offering. Components the team still operates alone are listed publicly at komms.app/docs.
- Company equity, alpha tokens, and emissions are legally and economically distinct.
- This paper seeks technical criticism, not formal endowment, allocation, or endorsement from any foundation.

Document discipline

The architecture, commercial model, and evidence register should be read together. A product definition describes the intended surface; it does not by itself establish production readiness, customer revenue, or mainnet economics.

12. Conclusion

A narrow commodity with broad demand

THE PROPOSAL Make Kasper-ordered data available, queryable, and provable under adversarial incentives, then sell that capability, first through Messaging and later through RPC, Data Availability, and ZK Proving.

Komms aims to become infrastructure for human teams and on-chain agents that need coordination, observation, memory, and proofs from the same verifiable fabric.

komms.app • komms.app/docs • info@komms.app

