$OPENBOOKS · CONCEPT MEMO

A market that produces a model

Economically weighted parallel inference — where expertise is decomposed by word, routing is decided by payment rather than a gating network, and compute follows demand automatically.

Status — concept. Partially built; nothing here is a claim about a shipped product.

Purpose — to test whether the idea is coherent, not to sell it.

Date — 16 August 2026

The claim, in one paragraph

Language models are trained once and then rented uniformly: the same weights answer every question, and the cost of an answer is unrelated to how much anyone wanted it. The alternative proposed here is to decompose expertise **by word**. Each word owns a public, append-only corpus of everything written using it. Each word has an agent whose only job is to keep that word's working definition current. Each word has a wallet. Asking a word a question pays that word, and the payment funds the answer. Words nobody asks about cost nothing and do nothing. The result is a system where compute is allocated by revealed demand rather than by a scheduler, and where the knowledge that differentiates it is contributed by the people who use it — and owned by them.

The mechanism

Four objects, and nothing else is required.

1. **The word.** A `$ticker` — a claimable, canonical symbol. `$PINK`, `$INSURANCE`, `$LIQUIDITY`. A word with two genuine senses becomes an index with children, so the unit is really a *sense*, not a string.
2. **The corpus.** Every post that used the word. Public, append-only, timestamped, cryptographically attributed to its author. This is the only asset in the system that cannot be copied cheaply, because it has to be written.
3. **The agent.** A process that reads the corpus and maintains the word's current working definition — not what a dictionary says, but what the people using it have made it mean. It re-reads only when the corpus has actually grown.
4. **The wallet.** An address belonging to the word. Invocations pay into it; replies and re-derivations are paid out of it. A word's balance is a public measure of how much that word is used.

> **THE LOAD-BEARING PROPERTY**
>
> Every unit of compute the system spends requires a prior payment from someone who wanted it. There is no scheduler, no idle fleet, and no way for cost to run ahead of revenue. This is unusual: most inference businesses provision capacity and hope for utilisation.

Why this is not a mixture of experts

The comparison is the obvious one and the differences are the whole point.

	Mixture of experts	This
Routing decided by	a learned gating network	**payment**
Expert boundary	emergent, opaque	one word — human-legible
Expert knowledge	frozen at training time	append-only, live, dated
Compute per expert	uniform	proportional to demand for it
Adding an expert	retrain	claim a word — fractions of a penny
Who owns the expertise	the model owner	whoever wrote the corpus
Why it answers as it does	not recoverable	readable, per contribution

The base model is a commodity input here, not the product. Swap it and the system still works; delete the corpora and there is nothing left. That inversion is what makes the economics behave differently from an AI company's.

Cost structure

These are derived from the constants actually in the codebase — context window, sample size, output caps — priced at published rates for a small fast model.

Operation	What it does	Sats	Marginal cost
Claim a word (batched)	mints the word, one-off	~11	$0.0000013
Claim a word (single tx)	same, unbatched	113	$0.000013
Answer an invocation	~1,700 in / 225 out	~26,000	$0.003
Re-derive a definition	~4,600 in / 300 out	~52,000	$0.006
Hold a word nobody uses	**no timer, no polling**	**0**	**$0.00**

Satoshis at $11.62/BSV, verified against a live miner policy. Model costs at published rates for a small fast model, sized from the context windows and output caps actually in the code.

> **THE RATIO THAT DECIDES EVERYTHING**
>
> A reply costs **~26,000 satoshis of compute against ~113 satoshis of chain**. Inference is roughly **230× the on-chain cost**, which means the blockchain is rounding error in this system's economics and the model bill *is* the business. Every instinct to optimise transaction fees is misdirected effort.
>
> The corollary is that inventory is close to unpriced: a namespace of **100,000 words costs about 13 cents** to mint. Capital is not needed to acquire words. It is needed to acquire corpus.

Fixed costs are approximately zero

There is no training run, no GPU fleet, no capacity to provision. Dormancy is free by construction: a word is only re-read when its corpus has grown by a threshold, so ten thousand unused words and ten unused words cost the same — nothing. The system's total spend is bounded by its own tick rate, not by the size of its inventory.

> **THE ONE UNFUNDED LINE — AND ITS FIX**
>
> Answering an invocation is paid for by the invoker. **Re-deriving a definition is not paid for by anyone.** It is the only operation whose cost scales with the platform's own activity rather than with demand, and left alone it is the line item that could run away.
>
> The fix follows from the design rather than fighting it: a word re-derives **out of its own wallet**, funded by the invocation fees it has already collected. Words that get used can afford to think about themselves; words that do not, do not. Platform exposure returns to zero.

Revenue structure

There is exactly one lever: **the price of an invocation**. Everything else is arithmetic.

Answering costs $0.003, so an invocation must clear roughly $0.005 to carry a working margin — around **43,000 satoshis**. That is nearly four hundred times what a post costs, which is the practical consequence of the ratio above: **asking a word a question and telling a word something are different products at different prices**, and pricing them alike would subsidise every question out of nothing. Gross margin is not a function of scale; it is a policy choice, set on the first day.

What caps that price is not competition from another board. It is the question *"would I rather just ask a general model?"* — and that ceiling is **low for common words and high for contested ones**. This single fact determines which words are worth owning, and it points the opposite way from intuition.

Words	Invocations / word / day	Revenue / yr	Compute / yr	Gross / yr
1,000	2	$3,650	$2,190	$1,460
10,000	5	$91,250	$54,750	$36,500
100,000	10	$1,825,000	$1,095,000	$730,000

Modelled, not measured. Invocation volume is the assumption doing all the work here — the unit costs are grounded, the demand is not. A 40% gross margin at every scale is the signature of a system with no fixed costs: it neither benefits from operating leverage nor suffers from its absence.

What has to be true

Three conditions. The first two are satisfied; the third is the whole risk.

1. **Dormancy must be free.** Otherwise inventory becomes a liability and the model inverts. *Satisfied* — re-derivation is gated on corpus growth, not elapsed time.
2. **Cost must not be able to precede revenue.** Otherwise volume becomes dangerous. *Satisfied* — corpus growth requires paid posts, so agent spend is downstream of payment by construction.
3. **For some words, the corpus must beat the base model.** This is the only question that matters, and it is not settled.

On the third condition

An agent tending `$INSURANCE` has no advantage over a general model. The concept is exhaustively covered in every training corpus, and a per-word agent adds latency and cost to produce a worse answer. The words where a live corpus genuinely wins are those whose meaning is **recent, contested, local, or moving** — emerging jargon, industry terms with a private sense, brand names, anything currently being argued over. Nobody had fixed the meaning of "cloud" in 2005.

This cuts directly against the instinct to buy the most commercially valuable keywords. Those are precisely the words where the corpus adds least. Two theories of value are easily confused:

- **Land grab** — own `$MORTGAGE` because keyword auctions price it highly. A legitimate speculation, but it is a speculation on namespace scarcity, not on the product working.
- **Product thesis** — own the words whose meaning is unsettled, because that is where a live corpus is the only source of a correct answer.

They imply different shopping lists, and conflating them would produce an expensive inventory of words that answer no better than a chatbot.

What kills it

- **Cold start.** A word with an empty corpus is strictly worse than a general model — slower, costlier, and less informed. The product does not exist until the corpus does, and the corpus is written by people who currently have no reason to write it. *This is the real capital requirement.*
- **Corpus capture.** Meaning is derived from usage, so a coordinated group that pays to post can make a word mean what it wants. This is simultaneously the feature (contested meanings get recorded honestly) and the attack (buy the definition of a competitor's brand). Any defence that involves editorial judgement contradicts the platform's stated ethos; any defence that does not must be economic.

- **Squatting.** If ownership follows whoever typed a word first, the valuable half of the namespace is claimable for pennies by a script. Ownership must track *share of corpus* instead — which also happens to be the position that makes the whole thesis coherent.
- **Custody concentration.** Deriving every agent's key from one seed makes operations tractable and makes the operator able to sign as any word. On a platform whose proposition is provable authorship, that is a contradiction to be managed rather than ignored.
- **The base model improving.** Every capability increment in general models raises the bar the corpus must clear. The defensible words are the ones no amount of training data reaches, because they are about what happened last week among these specific people.

**Keyword auctions rent attention against words nobody is allowed to own.
This owns the words, and pays the people who gave them meaning.**

What exists today

Component	Status
Claimable word tokens with canonical parsing	Built, live
Public append-only corpus, cryptographically attributed	Built, live — ~2,100 records anchored
Paid posting, on-chain, author-funded	Built, live
Per-word agent maintaining a live definition	Built — running on a handful of words
Definition anchored to an external prior	Built
Invocation as payment (agents self-funding)	Designed, not built
Key derivation for agents at scale	Designed, not built
Ownership tracking share of corpus	Open problem
Batch minting	Not built

Where capital would actually go

Not to compute, and not to inventory — both are close to free. The honest use of funds is:

- **Corpus acquisition.** Paying people to write the first several million words of usage, in a deliberately narrow domain where meaning is genuinely contested. This is the only line item that buys the thing competitors cannot copy.
- **Custody infrastructure.** Real key management, because the current arrangement is adequate for two agents holding pennies and inadequate for a namespace.

- **Engineering.** Batch minting, the self-funding loop, and ownership-by-contribution — the three unbuilt pieces on which the model depends.

The order matters. Building the mechanism before the corpus produces a machine that runs correctly and says nothing worth paying for.

> **ASSESSMENT**
>
> The economic shape is sound and unusually safe: no fixed costs, free dormancy, and a structural guarantee that spending cannot outrun income. It is very difficult for this system to lose money at scale.
>
> It is not difficult for it to be *worthless* at scale. The entire risk sits in one place — whether a live, owned, public corpus produces better answers than a general model for a class of words large enough to matter. That is a question about content, not architecture, and it is answerable cheaply: pick one domain where meaning is genuinely contested, buy the corpus, and measure whether anyone prefers the answer.