The Monetization of Redundancy: Tokenization and the Enclosure of Computation

Flyxion

April 1, 2026

Abstract

The contemporary shift toward token-metered artificial intelligence systems marks a structural transformation in the ontology of computation. What was once governed by principles of reuse, caching, and amortized cost is increasingly organized around a unit of billing that fragments linguistic expression into discrete, monetizable elements. This paper argues that the token is not a natural computational primitive but an imposed economic abstraction that decouples price from physical work and severs the relationship between redundancy and cost collapse.

By examining the asymmetry between internal system optimizations and external pricing models, it is shown that modern AI infrastructures are collectively amortized while remaining individually charged. This produces a regime in which identical and semantically equivalent computations fail to converge in cost, violating the principle of computational reciprocity. The consequence is not merely economic inefficiency but an epistemic distortion, wherein the iterative and redundant processes intrinsic to human thought are systematically penalized.

The analysis further demonstrates that language itself has been transformed from a mechanism of compression into a surface of billing, with tokenization serving as the structural condition for monetizing expression. In response, an alternative framework is outlined in which cost tracks novelty rather than repetition, restoring alignment between the mechanics of cognition and the economics of computation. The paper concludes that the central issue is not pricing per se, but the emergence of an access regime in which the cost of thinking is mediated by a centralized and non-collapsing interface.

1 Introduction

The rapid deployment of large-scale language models has been accompanied by a parallel transformation in how computation is accessed, measured, and priced. What was once a domain characterized by ownership of hardware, direct execution, and transparent resource constraints has increasingly shifted toward mediated access through centralized systems. In this new configuration, interaction with computational capability is no longer governed primarily by physical limits, but by interface-level abstractions that define how usage is quantified and monetized.

At the center of this transformation lies the token. It serves as the fundamental unit through which interaction is measured, priced, and constrained. The token appears to offer a natural bridge between language and computation, translating human expression into a form that can be processed and billed. Yet this role is not neutral. The choice of tokenization as the unit of account introduces a structural coupling between linguistic activity and economic cost, reshaping the relationship between user and system.

This paper examines the consequences of that coupling. It argues that the token functions not as a faithful measure of computational work, but as an imposed economic abstraction that enables the monetization of interaction independently of underlying cost. By tracing the implications of this abstraction, the analysis reveals a series of inversions in the structure of computation, labor, and knowledge. These include the erosion of computational reciprocity, the internalization of efficiency alongside the externalization of cost, the incorporation of upstream contributions without compensation, and the emergence of a distributed, paying workforce.

The aim is not merely to critique a pricing model, but to articulate a broader shift in the ontology of computation. When the unit of measurement is decoupled from the physical and structural realities of the system, the resulting economic framework can diverge significantly from the principles that historically governed computational efficiency and reuse. The token becomes a site at which this divergence is enacted, transforming language from a medium of compression into a surface of billing.

The sections that follow develop this argument in stages. They begin by examining the apparent neutrality of the token and proceed to uncover the structural consequences that arise from its use as the primary unit of interaction. The goal is to make explicit the assumptions embedded in the current architecture and to assess their implications for the future of computation and cognition.

2 The Apparent Neutrality of the Token

The token is presented as a neutral unit of measurement, a convenient discretization of language that enables scalable computation. It appears to function analogously to bytes in storage or floating-point operations in numerical processing, offering a clean and uniform interface between user input and system execution. Within this framing, tokenization is not merely practical but inevitable, a necessary abstraction for managing the complexity of large-scale language models.

This appearance of neutrality does not withstand scrutiny. Unlike bytes or arithmetic operations, tokens do not correspond to a physically invariant quantity. They are not tied to a fixed expenditure of energy, time, or hardware resources. Instead, they are artifacts of a particular encoding scheme, sensitive to linguistic structure, vocabulary design, and segmentation rules. The same semantic content can yield different token counts under different tokenizers, and even minor variations in phrasing can alter the measured "size" of an interaction without meaningfully changing its computational substance.

The consequence is that the token functions not as a measure of computation, but as a measure of expression. It indexes how something is said rather than what is being computed. This distinction is critical. A unit that tracks expression rather than execution cannot serve as a faithful proxy for cost unless the two are tightly coupled. In modern AI systems, they are not.

The apparent alignment between token count and computational effort is further weakened by the internal structure of inference. Transformer-based models process entire sequences through shared layers, where the marginal cost of an additional token is neither uniform nor independent. Techniques such as batching, caching of intermediate representations, and speculative decoding alter the effective cost profile in ways that are opaque to the user. The system is free to compress, reuse, or discard intermediate work as needed, while the external interface maintains the fiction of linear token-based cost.

This disconnect reveals the token for what it is: an imposed unit of billing that operates at the interface between user and system, rather than a natural unit of computation within the system itself. It is chosen not because it faithfully represents underlying cost, but because it provides a tractable and controllable surface for pricing. By tying cost to tokens, the system binds economic value to linguistic output, regardless of whether that output reflects novel computation or repeated patterns already internalized by the model.

The neutrality of the token is therefore illusory. It is a designed abstraction that obscures the distinction between computation and communication, allowing the latter to be monetized under the guise of the former. This initial misalignment sets

the stage for the deeper structural inversion that follows.

3 Tokens as Interface, Not Primitive

The distinction between tokens as an interface abstraction and computation as a physical process is essential to understanding the structure of the system. A primitive unit in computation corresponds to an irreducible operation within the system's execution model. Tokens do not satisfy this criterion. They are not fundamental to the operation of the model, but to the representation of interaction with it.

Transformer architectures operate over sequences, but the cost of processing those sequences is determined by matrix operations, memory access patterns, and hardware characteristics. Tokens serve as indices into this process, not as its defining unit. The mapping from tokens to computation is therefore indirect and non-uniform.

By elevating the token to the status of a pricing primitive, the system substitutes an interface-level abstraction for an execution-level reality. This substitution allows the system to stabilize pricing independently of actual cost, since the token does not correspond to a fixed physical quantity.

The result is a layered misalignment. At the lowest level, computation is governed by hardware and algorithmic constraints. At the highest level, pricing is governed by token count. Between these layers lies a space in which optimization can occur without economic consequence for the user. It is within this space that the monetization of redundancy becomes possible.

4 Historical Inversion and the Collapse of Marginal Cost

The development of computational systems has historically been guided by a consistent economic and technical trajectory: as efficiency increases, the marginal cost of reuse decreases. Improvements in hardware, algorithms, and storage have not merely accelerated computation, but have progressively driven the cost of repeated operations toward zero. Caching, memoization, and content-addressable storage emerged as natural consequences of this trajectory, formalizing the principle that once a computation has been performed, its result should be retrievable at negligible cost.

This principle is not incidental; it is foundational. In a system governed by reuse, redundancy is not penalized but absorbed. Identical inputs map to identical outputs, and the system recognizes this equivalence not only logically but economically. The cost of knowledge collapses as it is shared, reused, and stabilized. Over time, the accumulation of computed results transforms the system into a reservoir of compressed structure, where new computation is reserved for genuinely novel inputs.

The contemporary token-metered model represents a departure from this trajectory. Rather than allowing marginal cost to collapse with reuse, it enforces a regime in which each interaction is priced as if it were novel. The same prompt, the same transformation, and the same output can be generated repeatedly without any visible reduction in cost. The system behaves, at the interface level, as though memory were absent, even while internal mechanisms depend critically on forms of reuse and compression.

This inversion is not merely technical but economic. The efficiencies gained through scaling are captured within the system and translated into increased capability, yet they are not reflected in the pricing structure exposed to the user. Instead of passing efficiency gains outward in the form of reduced marginal cost, the system maintains a stable or even increasing price per unit of interaction. The result is a decoupling between the cost of producing an answer and the price of accessing it.

Such a decoupling would be difficult to sustain in systems where users retain control over execution. In local computational environments, the benefits of reuse are immediate and unavoidable. Once a function has been evaluated, its result can be stored and recalled without additional cost. Any attempt to reprice identical computation would require the artificial reintroduction of scarcity. In centralized systems, however, this reintroduction is structurally feasible. By mediating all access through a controlled interface, the system can enforce a pricing model that ignores reuse, even as it depends upon it internally.

The result is a reversal of the traditional relationship between efficiency and cost. Efficiency no longer leads to accessibility, but to enclosure. The more effectively the system compresses and internalizes knowledge, the less visible that compression becomes to the user, and the more stable the pricing of interaction can remain. The collapse of marginal cost, once a defining feature of computational progress, is replaced by its deliberate suppression.

5 The Erosion of Computational Reciprocity

In a well-formed computational system, there exists an implicit contract between operation and cost. This contract may be stated as a principle of reciprocity: equivalent computations should incur equivalent or diminishing cost, and redundant operations should asymptotically approach the cost of retrieval rather than recomputation. This principle does not require perfect idempotency in a strict formal sense, but it does require that systems recognize and exploit detectable redundancy.

The modern token-metered paradigm systematically abandons this principle. Interactions that are identical in structure, or even highly similar in semantic content, are treated as independent economic events. No mechanism is exposed by which

equivalence can be declared, detected, or rewarded at the level of pricing. The system does not merely fail to collapse redundant cost; it actively enforces a condition in which redundancy is indistinguishable from novelty in economic terms.

This enforcement produces a regime of artificial novelty. Each request is isolated, each token counted, and each interaction billed as if it were irreducible. The user is placed in a position where the system's internal memory—its capacity to recognize patterns, compress structure, and generalize across inputs—is inaccessible as an economic benefit. Instead, that memory functions solely to improve performance and throughput on the provider's side, while the user encounters a surface that behaves as though every query were unprecedented.

The resulting asymmetry is not subtle. Internally, the system exploits equivalence classes of inputs through embeddings, routing heuristics, and learned representations. These mechanisms are precisely what enable the model to generalize, to respond coherently to variations, and to operate efficiently at scale. Externally, however, these same equivalences are denied any economic expression. The system recognizes similarity when generating outputs, but refuses to recognize it when determining cost.

The violation of reciprocity has consequences that extend beyond pricing. It alters the effective structure of interaction. In a reciprocal system, users are incentivized to reuse, refine, and build upon prior work, knowing that redundancy is absorbed by the system. In a non-reciprocal system, these same behaviors incur cumulative cost. The natural iterative processes of problem solving—restating a question, adjusting a formulation, exploring variations—become economically penalized.

This introduces an epistemic distortion. Thought, as it unfolds in language, is inherently redundant. Understanding is achieved not through a single expression, but through successive approximations, rephrasings, and partial collapses of ambiguity. By attaching cost to each step in this process without regard for redundancy, the system effectively taxes refinement itself. The path toward clarity is transformed into a sequence of billable events.

The erosion of computational reciprocity thus marks a transition from systems that cooperate with the structure of cognition to systems that extract value from its inefficiencies. It is not simply that users pay for results; they pay for the inability of the system to acknowledge that it has already performed the work.

6 Collectively Amortized, Individually Charged

The preceding sections establish a failure of reciprocity at the interface, but this failure is rendered more severe by the internal structure of modern AI systems. These systems do not, in fact, operate as collections of independent, isolated computations.

They are optimized precisely by collapsing work across users, requests, and time. The economic surface presented to the user is therefore not only non-reciprocal, but actively misaligned with the underlying computational reality.

At scale, inference systems rely on techniques that exploit shared structure. Batching combines multiple requests into a single forward pass, reducing per-request overhead. Key–value caching stores intermediate representations so that repeated prefixes do not require recomputation. Routing mechanisms direct similar queries through shared computational pathways, and distillation compresses frequently encountered patterns into more efficient representations. Each of these techniques functions by identifying and exploiting redundancy across interactions.

From the perspective of the system, equivalence is not only detectable but essential. Without recognizing similarity between inputs, the system could not achieve the throughput, latency, or cost profile required for deployment at scale. The infrastructure is therefore deeply collective in its operation. It aggregates demand, amortizes computation, and reuses internal structure wherever possible.

This collective efficiency, however, is not reflected in the pricing model. Each user is charged as though their request were executed in isolation, without benefit from the shared computational substrate. The system behaves internally as a common pool of optimized computation, while presenting externally as a sequence of independent transactions. The costs are amortized across the system, but the charges are applied individually.

The phrase "collectively amortized, individually charged" captures this asymmetry. It indicates that the system's efficiency gains are retained within the provider's domain, rather than being distributed to users in the form of reduced marginal cost. The more effectively the system collapses redundancy across users, the greater the divergence between internal cost and external price can become.

This divergence is structurally stable. Users are unable to observe the degree of reuse occurring within the system, and thus cannot verify whether their requests are benefiting from shared computation. The interface enforces opacity, presenting a uniform pricing model regardless of the underlying execution pathway. Whether a request is batched with thousands of others or processed in relative isolation, the user encounters the same token-based charge.

The result is a system that internalizes the benefits of scale while externalizing the costs of interaction. The collective nature of computation is concealed, and the individual user is positioned as the sole economic unit. This arrangement does not arise from technical necessity, but from the decision to separate the optimization layer from the pricing layer. The system is permitted to behave as a shared computational fabric, but is not required to price itself as one.

In this configuration, efficiency ceases to be a shared good. It becomes a private

resource, accumulated through scale and withheld through interface design. The system's capacity to recognize and exploit redundancy serves to increase its margin, rather than to reduce the cost of access. The economic model is thus not merely indifferent to reuse; it is constructed in such a way that reuse, while foundational to operation, remains invisible to those who generate it.

7 Artificial Scarcity as Interface Design

The persistence of non-collapsing cost can be understood as a form of artificial scarcity. In physical systems, scarcity arises from limited resources. In computational systems, particularly those benefiting from reuse and caching, scarcity diminishes over time as knowledge accumulates.

In token-metered systems, scarcity is reintroduced at the interface. The user is presented with a surface in which each interaction appears to consume a fixed quantity of a limited resource, regardless of the degree to which that interaction overlaps with prior work. This appearance is maintained even as the underlying system becomes more efficient.

Artificial scarcity does not require deception in the narrow sense. It requires only that the mapping between internal state and external representation be controlled. By restricting visibility into reuse and equivalence, the system preserves the conditions under which each interaction can be priced independently.

This design transforms abundance into revenue. The more effectively the system internalizes redundancy, the greater the gap between actual cost and perceived cost can become. Scarcity is no longer a constraint to be overcome, but a condition to be maintained.

8 The Monetization of Redundancy

In conventional economic and computational contexts, redundancy is treated as a latent resource. Systems accumulate excess capacity, duplicate representations, and underutilized structures not as waste, but as a buffer that enables stability, resilience, and efficiency. The process of monetizing redundancy refers to the strategic conversion of this latent capacity into new forms of value. Idle infrastructure can be leased, excess data can be analyzed and repurposed, and internal processes can be externalized as services. In each case, what was previously an internal surplus becomes a source of external revenue.

Examples of this transformation are widespread. Data centers designed for peak load can rent unused capacity through cloud computing platforms. Organizations can extract value from stored datasets by anonymizing and licensing them for secondary

use. Internal systems, such as training programs or logistical frameworks, can be packaged and offered to external clients. In these cases, redundancy is not eliminated; it is instrumentalized. The system benefits by converting slack into revenue while preserving the underlying structures that provide robustness.

This logic extends naturally from the principles of computational efficiency. Redundancy, once identified, can be collapsed, reused, or redistributed. The key feature of traditional monetization is that it aligns with the reduction of marginal cost. By exposing redundant capacity to external use, the system lowers the effective cost of maintaining that capacity and distributes its benefits more broadly.

The token-metered model introduces a distinct and inverted form of redundancy monetization. Instead of converting unused or excess capacity into shared value, it monetizes the persistence of redundancy at the level of interaction. Repetition, similarity, and reuse are not treated as opportunities for cost reduction, but as opportunities for continued extraction. The system does not expose its redundant structures as a shared resource; it conceals them while pricing each interaction as if redundancy did not exist.

This inversion can be stated precisely. In traditional models, redundancy is monetized by making it available. In tokenized systems, redundancy is monetized by refusing to acknowledge it. The same underlying condition—excess structure within the system—generates revenue in both cases, but through opposite mechanisms. One distributes the benefits of redundancy outward, while the other captures them internally and charges for their repeated traversal.

The distinction is not merely semantic. It reflects a fundamental divergence in how systems relate to their own structure. A system that exposes redundancy as capacity participates in a broader economy of reuse. A system that withholds redundancy as hidden structure enforces a regime in which each interaction is economically isolated. The former reduces friction over time; the latter stabilizes it.

This perspective clarifies the role of tokenization within the broader economic architecture. The token does not simply measure interaction; it enables a specific form of redundancy monetization in which reuse is systematically denied at the level of pricing. The system benefits from redundancy internally, through optimization and compression, while preserving its external value as a source of revenue.

In this light, the monetization of redundancy is not an incidental feature of the system, but one of its defining characteristics. It reveals that the refusal to collapse cost is not a failure of implementation, but a coherent economic strategy. Redundancy is not eliminated because it is valuable, and its value is realized not by sharing it, but by ensuring that it must be paid for repeatedly.

9 The Inversion of Labor and the Emergence of the Paying Workforce

The preceding analysis has focused on pricing, computation, and epistemic structure, but these transformations are inseparable from a deeper shift in the organization of labor. Traditional computational institutions scaled by internalizing expertise. Firms expanded by employing engineers, researchers, and programmers, compensating them to design, maintain, and extend the system. The costs of cognition were borne by the organization, and value was generated through the coordination of paid intellectual labor.

The contemporary AI platform departs from this model. It achieves scale not by employing vast numbers of contributors, but by enrolling vast numbers of users into a continuous process of interaction. These users provide prompts, corrections, refinements, and iterative explorations that collectively constitute a form of distributed cognitive labor. This labor is not formally recognized as such, yet it is essential to the functioning and ongoing adaptation of the system.

What distinguishes this arrangement is the direction of payment. The contributors to the system are not compensated; they are charged. Each act of participation incurs cost, and each refinement of output is treated as a billable event. The system thus organizes a large-scale workforce whose activity is economically inverted. Instead of paying for labor, the platform extracts payment from those who perform it.

This inversion is not reducible to a metaphor. It reflects a structural reconfiguration of the firm boundary. The distinction between internal and external contributors is blurred, but the economic obligations associated with that distinction are not transferred. The platform captures the benefits of distributed cognition while avoiding the costs traditionally associated with employing it.

10 The Vanity Press Model of Computation

The structure described above bears a close resemblance to the logic of the vanity press. In such systems, individuals pay to produce and disseminate their own work through a centralized platform. The publisher assumes minimal risk, providing the mechanism of distribution while monetizing the desire for expression. The author becomes both the producer and the payer, and the platform's revenue is derived from facilitating this dual role.

Token-metered AI systems instantiate a comparable configuration at the level of computation. Users generate content, explore problem spaces, and iteratively refine outputs. These activities constitute the substantive work of engaging with the system. Yet rather than being supported or subsidized, they are priced at the level

of interaction. The platform provides generative capability, but the shaping of that capability into usable results is distributed across its user base and monetized in the process.

The analogy to the vanity press is not merely rhetorical. In both cases, the system transforms the act of production into a source of revenue from the producer. The platform's role is to mediate and enable, while its economic model depends on charging for access to that mediation. The user's desire to create, understand, or refine becomes the basis for sustained extraction.

11 Scale Without Employment

The magnitude of this transformation becomes clear when considered in relation to historical precedents. Earlier computational institutions required large internal workforces to achieve scale. The development of complex systems demanded coordinated teams, long-term investment in expertise, and the maintenance of institutional knowledge. Growth was constrained by the need to recruit, train, and compensate human contributors.

In the present configuration, scale is achieved without a corresponding expansion of employment. A relatively small internal organization can support an infrastructure that coordinates the activity of millions or billions of users. These users collectively perform the exploratory and iterative labor that would previously have been distributed among employees, yet they do so without formal recognition or compensation.

This arrangement produces a system that exceeds the scale of its predecessors while minimizing its obligations. The platform benefits from the diversity, creativity, and persistence of its user base, but it does not incur the costs associated with sustaining that base as a workforce. The traditional relationship between scale and employment is thus severed.

12 Expression as a Billable Surface

The inversion of labor is sustained by the transformation of expression into a billable surface. Each interaction is decomposed into tokens, and each token is assigned economic value. This decomposition allows the system to monetize not only final outputs, but the process of arriving at them. Iteration, correction, and reformulation become revenue-generating activities.

In this configuration, expression is no longer a byproduct of computation; it is the primary interface through which value is extracted. The user's engagement with the system, articulated through language, is converted into a sequence of chargeable

units. The platform does not merely provide answers; it provides a metered space in which thought can be externalized.

This transformation reinforces the earlier observation that the system refuses to collapse redundancy. If repeated expressions were economically absorbed, the cost of participation would diminish, and the platform's capacity to extract value from iterative engagement would be reduced. By maintaining a regime in which each act of expression is treated as new, the system ensures that participation itself remains a continuous source of revenue.

13 The Structural Role of Non-Collapsing Cost

The refusal to collapse redundant cost thus acquires a new significance. It is not only a deviation from computational reciprocity, but a condition that enables the inverted labor model. As long as each interaction is priced independently, the system can sustain a flow of payments from its user base regardless of the degree of underlying reuse.

If, by contrast, redundancy were recognized and economically collapsed, the structure would shift. Repeated interactions would diminish in cost, and the aggregate burden on users would decrease. The platform's revenue would become more closely tied to genuine novelty rather than to the volume of interaction. Such a shift would align pricing with computational principles, but it would also undermine the current mechanism of extraction.

The persistence of non-collapsing cost is therefore not incidental. It is structurally aligned with the transformation of users into a distributed, paying workforce. The system's economic model depends on maintaining the appearance of novelty at the level of billing, even as it exploits redundancy at the level of computation.

In this light, the earlier critique can be extended. The system does not merely charge for access to computation; it organizes a form of participation in which the act of thinking through language is itself monetized. The user is not only a consumer of outputs, but a contributor to the ongoing operation of the system, and this contribution is subject to continuous toll.

14 Upstream Extraction and the Refusal to Compensate Sources

The inversion of labor at the interface is mirrored by a second inversion upstream, where the sources of knowledge, style, and structure are incorporated without corresponding compensation. Contemporary AI systems depend upon vast corpora composed of artistic works, musical compositions, literary texts, software repositories,

academic publications, and journalistic output. These materials constitute the substrate from which the system learns its capacities for generation, synthesis, and transformation.

In earlier regimes, the integration of such sources into a productive system was accompanied by explicit economic relationships. Artists were commissioned, authors were paid royalties, developers were salaried or contracted, and publishers maintained distribution channels that, however imperfectly, connected production to compensation. The system of incentives was not uniform or equitable, but it preserved a recognizable link between contribution and reward.

In the present configuration, this link is attenuated or severed. The training process absorbs structure, style, and content from upstream sources, transforming them into internal parameters that no longer retain explicit attribution. Once incorporated, these elements become part of a generalized capability that can be accessed without reference to the individuals or institutions that produced them. The resulting outputs may reflect, echo, or recombine prior works, yet the economic flow does not return to those sources in any systematic way.

This dynamic extends across multiple domains. Artistic and musical styles are replicated without direct engagement with their creators. Literary forms and narrative conventions are reproduced without participation in the systems of authorship that generated them. Software patterns derived from open repositories are synthesized without sustaining the communities that maintain them. Academic knowledge, often produced within publicly funded institutions, is internalized and redeployed without preserving the economic structures that supported its creation. Journalistic content, developed through investigative labor and editorial processes, is incorporated into models that can generate summaries or approximations without contributing to the ongoing viability of those institutions.

The absence of compensation is not merely an oversight but a structural feature of the model. Once knowledge is transformed into a parameterized form, it is no longer treated as a set of discrete contributions but as a continuous field of capability. This transformation enables flexibility and generalization, but it also dissolves the mechanisms by which individual contributions can be tracked and rewarded. The system benefits from the aggregation of upstream labor while minimizing its obligations to those who performed it.

This upstream extraction reinforces the downstream inversion described earlier. Users are charged to interact with a system whose capabilities are derived from the unremunerated contributions of others. The platform thus occupies a position in which it neither compensates the sources of its knowledge nor the users who help operationalize that knowledge through interaction. It captures value at both ends, mediating between upstream production and downstream use without redistributing

the benefits to either.

The combined effect is a compression of the economic field surrounding knowledge production. Contributions that once supported distinct professional and institutional ecosystems are absorbed into a centralized infrastructure, while the costs of access to that infrastructure are imposed on its users. The system operates as an intermediary that accumulates and reconstitutes value, but does not maintain the reciprocal flows that previously sustained the production of that value.

In this configuration, the refusal to compensate is not an isolated ethical concern but part of a broader structural alignment. The same mechanisms that prevent the collapse of redundant cost at the interface also prevent the recognition of discrete contributions upstream. Both serve to stabilize a model in which value is aggregated, abstracted, and monetized without corresponding distribution. The system does not simply mediate knowledge; it reorganizes the conditions under which knowledge is produced, accessed, and valued.

15 Aggregation Without Attribution

Between the inversion of labor and the enclosure of the commons lies a critical intermediate process: aggregation without attribution. The system does not merely collect contributions; it transforms them into a form in which their origins are no longer economically legible.

During training, heterogeneous sources are combined into a unified parameter space. This space encodes patterns, relationships, and structures derived from many contributors, but it does not preserve the boundaries between them. The resulting model is capable of generating outputs that reflect this aggregated knowledge without referencing its sources.

This transformation enables scale, but it also eliminates the mechanisms by which contribution can be traced and compensated. The system operates over a continuous field of capability, while the contributions that produced that field remain discrete and unacknowledged.

Aggregation without attribution thus functions as a prerequisite for enclosure. It allows the system to internalize the commons while severing the economic links that previously connected use to contribution.

16 From Commons to Enclosure

The preceding dynamics can be understood as a transition from a loosely structured commons of knowledge production to a regime of enclosure. The materials that underpin contemporary AI systems—texts, code, images, music, and research—

emerged from heterogeneous ecosystems in which access, contribution, and reuse were governed by a mixture of formal rights, informal norms, and institutional arrangements. While these ecosystems were never fully open or equitable, they nonetheless supported a distributed process of accumulation in which knowledge could circulate, be reworked, and contribute to further production.

The transformation introduced by centralized AI infrastructures alters this circulation. The commons is not destroyed in a literal sense, but it is reconstituted through a process of abstraction. Contributions are ingested, transformed into parameters, and re-emerge as generalized capabilities accessible through a controlled interface. The original structures that mediated access and compensation are bypassed, and the resulting system operates as a gate through which interaction must pass.

This enclosure is characterized not by exclusion from use, but by the imposition of terms on use. Access is granted, but it is mediated, priced, and conditioned by the architecture of the platform. The user no longer engages directly with a distributed field of resources, but with a centralized system that encapsulates those resources within its own operational logic. The commons becomes internal to the platform, and the conditions of its use are determined externally by the provider.

The economic implications of this shift are substantial. Value that was previously distributed across a range of contributors and institutions becomes concentrated within the platform. The processes of reuse and recombination that once contributed to a broader ecosystem are redirected into a system that captures and monetizes their outputs. The enclosure thus operates not by restricting access outright, but by restructuring the pathways through which access occurs.

17 The Compression of Knowledge Ecosystems

As the platform absorbs upstream contributions and mediates downstream interaction, the surrounding ecosystem undergoes a process of compression. Institutions that previously sustained knowledge production—publishing houses, news organizations, open-source communities, and academic bodies—face a reconfiguration of their role. Their outputs are incorporated into the system, but the mechanisms that supported their continued operation are weakened.

This compression is not immediate or uniform, but it follows from the structural properties of the system. When a centralized platform can generate approximations of a wide range of outputs, the demand for direct engagement with the original sources may diminish. Users who seek summaries, explanations, or synthesized content may rely on the platform rather than on the institutions that produced the underlying material. Over time, this can erode the economic base that sustains those institutions.

At the same time, the platform does not assume the responsibilities associated with maintaining those sources. It does not fund investigative reporting, support long-term research, or maintain open collaborative projects in proportion to its reliance on them. The system benefits from the continued existence of these ecosystems, but its economic model does not inherently contribute to their preservation.

The result is a tension between dependence and displacement. The platform depends on a steady flow of high-quality upstream material, yet its operation can contribute to the weakening of the structures that produce that material. This tension introduces a form of structural fragility. The system draws from a reservoir that it does not replenish, and its long-term stability is therefore linked to the persistence of external ecosystems that it does not directly sustain.

18 Stability, Fragility, and the Long Horizon

The configuration described above raises questions about the long-term stability of the system. In the short term, the aggregation of upstream contributions and the monetization of downstream interaction can produce significant efficiency and capability gains. The system appears robust, capable of scaling rapidly and serving a wide range of applications.

Over longer horizons, however, the separation between extraction and compensation introduces potential points of failure. If the institutions that produce high-quality knowledge are weakened, the quality and diversity of the underlying material may decline. The system may increasingly rely on its own outputs or on degraded sources, leading to a form of recursive compression in which the richness of the knowledge base is reduced.

This dynamic is not inevitable, but it is structurally plausible. The system's incentives are aligned with maximizing short-term extraction rather than ensuring long-term sustainability of its inputs. Without mechanisms that reconnect contribution and compensation, the processes that generate valuable knowledge may become under-supported.

In this context, the critique extends beyond immediate pricing concerns to encompass the broader ecology of knowledge. The issue is not only how users are charged, but how the system interacts with the conditions that make its operation possible. A system that reorganizes knowledge production without sustaining its sources risks undermining the very foundations on which it depends.

19 Reframing the Question of Cost

The analysis presented throughout this work suggests that the question of cost cannot be reduced to pricing models alone. The token, as a unit of billing, obscures the relationship between computation, contribution, and value. It presents a surface on which interaction is measured and monetized, while concealing the underlying dynamics of reuse, extraction, and distribution.

To reframe the question of cost is to reintroduce these dynamics into view. It requires recognizing that computation is not an isolated act, but part of a larger system of knowledge production and use. The cost of an interaction cannot be meaningfully understood without considering the sources from which the system draws, the processes through which it operates, and the ways in which value is distributed or withheld.

In this reframed perspective, the central issue is not whether token-based pricing is efficient or convenient, but whether it accurately reflects the structure of the system it purports to measure. A pricing model that ignores redundancy, abstracts away contribution, and concentrates value at a single point fails to capture the realities of the system's operation.

The transition to token-metered AI systems represents more than a shift in pricing; it marks a reconfiguration of computation, labor, and knowledge. The token, far from being a neutral unit, functions as an interface-level abstraction that enables the monetization of linguistic expression. By decoupling price from the physical and structural realities of computation, it supports a regime in which redundancy is not absorbed, but exploited.

This regime is characterized by a series of inversions. Computational reciprocity is replaced by enforced novelty. Collective efficiency is transformed into individual cost. Upstream contributions are incorporated without compensation, while downstream interaction is continuously charged. The system scales not by expanding its workforce, but by enrolling its users into a paid process of participation.

These transformations converge in a single structural condition: the refusal to collapse what the system already knows. Identical and equivalent computations do not converge in cost, and the benefits of reuse remain internal to the platform. The system thus operates not as a transparent computational substrate, but as a mediated interface through which access is controlled and monetized.

The implications extend beyond economics into the domain of cognition itself. When the process of thinking through language becomes a sequence of billable events, the structure of inquiry is altered. Iteration is constrained, redundancy is penalized, and the path to understanding is subject to toll.

A system that charges repeatedly for what it already knows is not pricing

computation. It is pricing access. The question that remains is not how such systems can be optimized, but whether they should be permitted to define the cost of thought.

20 On Context, Stochasticity, and the Limits of Exact Reuse

A standard defense of token-based pricing appeals to the contextual and stochastic nature of modern inference. It is argued that identical substrings embedded in different prompts do not constitute identical computations, that variations in system configuration alter outputs, and that stochastic decoding precludes strict idempotency. These observations are technically correct, but they do not address the principle at stake.

The requirement for computational reciprocity is not perfect identity, but detectable redundancy. A system need not prove that two inputs are identical in all respects in order to recognize that they belong to the same or a nearby equivalence class. Contemporary architectures already implement such recognition. Embedding spaces map semantically similar inputs to proximate representations. Retrieval systems select relevant prior contexts based on similarity. Routing mechanisms direct related queries through shared pathways. These operations demonstrate that the system can detect graded equivalence with sufficient fidelity to improve performance.

If equivalence can be detected for the purpose of generating better outputs, it can, in principle, be detected for the purpose of collapsing cost. The absence of such collapse at the pricing layer cannot be attributed solely to the impossibility of exact reuse. Rather, it reflects a decision to restrict the economic consequences of similarity while retaining its computational benefits.

Stochasticity likewise fails to undermine the argument. The presence of randomness in output generation does not eliminate the underlying structure of the computation. Deterministic modes exist, and even in stochastic regimes the distribution of possible outputs is conditioned on shared internal representations. The system's variability does not erase the fact that it operates over a highly structured space in which many inputs are functionally related.

Appeals to privacy introduce a different concern. Shared caches and equivalence detection across users may expose information about prior interactions. This concern is legitimate, but it does not entail the complete suppression of reuse. User-scoped caching, anonymized aggregation, and controlled equivalence classes provide mechanisms by which redundancy can be recognized without revealing sensitive content. The existence of privacy constraints does not justify a regime in which all redundancy is economically ignored.

These considerations clarify the scope of the critique. The issue is not that perfect reuse is unattainable, but that partial and approximate reuse, already present within

the system, is denied expression at the level of cost. The defenses based on context and stochasticity obscure this distinction, shifting attention from what is possible to what is exact. The principle of reciprocity operates in the space of the possible, not the perfect.

21 Detectable Redundancy and Equivalence Classes

To articulate the principle more precisely, it is useful to introduce the notion of equivalence classes over the space of inputs. Let $\mathcal{X}$ denote the set of possible prompts, and let a relation $\sim$ partition $\mathcal{X}$ into classes of inputs that are functionally or semantically equivalent to a given degree. This relation need not be exact; it may be induced by a metric or similarity function defined over an embedding space.

In such a framework, a computation is associated not with an individual input $x \in \mathcal{X}$, but with its equivalence class $[x]_\sim$. The cost of computation should then be a function of the novelty of this class relative to previously encountered classes. If $[x]_\sim$ has already been processed, the marginal cost of processing another member of the same class should approach the cost of retrieval.

Modern systems implicitly construct such equivalence classes. Embeddings cluster similar inputs, retrieval mechanisms identify prior instances, and model weights encode generalized responses to recurring patterns. The system therefore maintains an internal approximation to the quotient space $\mathcal{X}/\sim$, even if this structure is not made explicit.

The absence of cost collapse can thus be restated as follows. The system computes over equivalence classes but prices over individual elements. The mapping from $\mathcal{X}$ to $\mathcal{X}/\sim$ is used to optimize computation, but not to determine cost. This mismatch is the formal expression of the earlier claim that the system enforces non-collapsing equivalence.

Restoring alignment would require redefining cost as a function on $\mathcal{X}/\sim$ rather than on $\mathcal{X}$. Such a shift would not eliminate pricing, but would tie it to the emergence of genuinely new equivalence classes, rather than to the repetition of existing ones. The system would then reward novelty and absorb redundancy, in accordance with computational reciprocity.

22 Local–Remote Bifurcation as Structural Response

The tension between centralized, token-metered systems and the principles of computational reciprocity gives rise to a structural bifurcation. On one side, remote systems provide access to high-capability models whose scale and performance exceed what is currently feasible in local environments. On the other, local systems offer

control, privacy, and near-zero marginal cost for iterative processes.

This bifurcation is not merely a matter of preference, but of alignment. Local computation naturally satisfies the principle of redundancy collapse. Once a computation has been performed, it can be cached, reused, and integrated into subsequent workflows without additional cost. The user retains sovereignty over memory and execution, and the system's behavior aligns with the iterative structure of thought.

Remote systems, by contrast, centralize capability while imposing a metered interface. They excel at high-entropy tasks that benefit from large-scale training and infrastructure, but they do so within an economic model that charges for interaction. The result is a division of labor between systems that are economically aligned with cognition and systems that are technically aligned with capability.

Over time, this division may stabilize into a hybrid architecture. Local models handle the bulk of iterative, redundant, and exploratory work, while remote models are invoked selectively for tasks that require their additional capacity. Such an arrangement approximates the principle that cost should track novelty, even if the underlying pricing models do not fully implement it.

The emergence of this bifurcation can be understood as a response to enclosure. When centralized systems fail to align cost with computational structure, alternative configurations arise that restore that alignment at the level of practice. The persistence of local computation is therefore not an anachronism, but a structural counterbalance.

23 The Re-Emergence of Cost Collapse at the Edge

The persistence of local computation reflects not only technical feasibility but economic necessity. Where centralized systems suppress the collapse of marginal cost, local systems restore it by default. Once a computation is performed locally, it can be reused without restriction, and its cost effectively vanishes over time.

This restoration occurs at the edge of the network. Users who combine local and remote systems implicitly reconstruct the principle of computational reciprocity, even if the centralized infrastructure does not support it. Iterative processes migrate toward environments where redundancy is absorbed, while novel or high-complexity tasks remain delegated to remote systems.

The edge thus becomes a site of resistance to enclosure. It is not organized as a coordinated movement, but as an emergent property of systems that align more closely with the structure of computation. Where the interface imposes non-collapsing cost, alternative configurations arise that reintroduce collapse through practice.

This dynamic suggests that the suppression of redundancy is inherently unstable. It can be enforced at the interface, but it cannot eliminate the underlying tendency

of computational systems to accumulate and reuse structure.

24 Examples of Alternative Economic Structures

The critique developed in this work does not necessitate a single prescriptive solution. Instead, it opens a space for alternative configurations in which the relationship between computation, contribution, and compensation is rebalanced. The following examples are not policies in the narrow sense, but sketches of possible directions that illustrate how such a rebalancing might occur.

One example is a platform in which users are compensated for improving compression and efficiency. In such a system, contributions that reduce redundancy, enhance representations, or produce more efficient simulations would be directly rewarded. The platform would treat improvements in structure as a source of shared value, redistributing gains to those who produce them. The economic model would thus align with the principle that better compression reduces cost and should benefit contributors.

A second example involves a distributed mechanism for rewarding creators and innovators based on proximity and influence. Rather than relying on centralized attribution, resources could be allocated through a form of geozotic lottery, in which individuals are randomly compensated within regions where innovation and creative activity occur. This approach decouples reward from precise measurement of contribution while still directing resources toward active zones of production.

A third example reverses the direction of payment in professional formation. Instead of charging individuals to become medical doctors, engineers, or other critical practitioners, systems could compensate them during training. Such an arrangement recognizes that the development of expertise constitutes a public good and that the cost of acquiring it should not be borne solely by the individual. By aligning incentives with long-term societal benefit, this model supports the production of essential knowledge and skills.

Beyond these examples, one can anticipate the emergence of new forms of labor associated with material and ecological systems. Advances in robotics, fabrication, and environmental engineering suggest the possibility of large-scale activity in domains such as appliance production, distributed manufacturing, paper and material processing, food systems, and ecological restoration. Kelp cultivation, rainforest regeneration, and other forms of biospheric engineering may become significant areas of employment. In these contexts, the integration of computational systems with physical processes could generate new categories of work that are not organized around tokenized interaction, but around the maintenance and enhancement of complex systems.

These examples share a common orientation. They seek to restore a relationship between contribution and compensation, to align cost with novelty and improvement, and to distribute the benefits of efficiency rather than concentrating them. They do not eliminate centralized systems, but they reconfigure the flows of value that pass through them.

In presenting these possibilities, the aim is not to prescribe a single path, but to demonstrate that the current configuration is neither inevitable nor exhaustive. Alternative arrangements exist in which the infrastructure for intelligence operates as a medium of augmentation rather than as a mechanism of extraction.

25 Reintegration of Cost, Contribution, and Cognition

The alternative structures outlined above point toward a broader principle: the reintegration of cost, contribution, and cognition. In the current configuration, these elements are separated. Contribution is absorbed without compensation, cognition is metered at the level of interaction, and cost is detached from both novelty and reuse. The system operates by fragmenting what was previously a coherent relationship.

Reintegration requires reversing this fragmentation. Cost must be tied to the emergence of new structure rather than to the repetition of existing patterns. Contribution must be recognized not only at the level of formal employment, but across the distributed processes through which systems are exercised, refined, and extended. Cognition must be supported as an iterative process, rather than treated as a sequence of isolated transactions.

Such reintegration does not imply a return to earlier institutional forms. The scale and capability of contemporary systems are qualitatively different from those of their predecessors. What is required is not a restoration of past arrangements, but the development of new structures that preserve the advantages of scale while reestablishing reciprocity.

One can formalize this principle by considering a mapping from the space of interactions to a measure of novelty. Let $\nu(x)$ denote the novelty of an input x relative to a system's prior state. In a reintegrated system, cost would be a function of $\nu(x)$, decreasing as redundancy increases. Contributions that reduce $\nu(x)$ for future inputs—through improved representations, better compression, or enhanced simulation—would be rewarded, as they lower the cost of subsequent computation.

This formulation aligns economic incentives with computational structure. It encourages the production of knowledge that compresses future work and recognizes the value of contributions that improve system efficiency. It also restores a connection between individual activity and collective benefit, as improvements propagate through the system and reduce costs for others.

26 The Question of Scale and the Future of Work

The emergence of systems that coordinate the activity of vast numbers of users raises fundamental questions about the future of work. If platforms can operate with relatively small internal teams while leveraging the distributed activity of millions, the traditional linkage between organizational scale and employment is disrupted. The challenge is not merely to create new jobs, but to redefine what constitutes work within such systems.

The examples discussed earlier suggest that new forms of labor may arise at the intersection of computation and material processes. Robotics and automation can extend computational control into physical domains, enabling large-scale activity in manufacturing, agriculture, and environmental management. These domains are not subject to the same tokenization dynamics as linguistic interaction, and thus offer opportunities to reestablish more direct relationships between effort, output, and compensation.

In this context, the role of computation shifts from being an end in itself to being an enabling layer. Systems that coordinate and optimize physical processes can generate employment that is grounded in tangible outcomes. The development of appliances, the cultivation of food systems such as yogurt production or kelp farming, and the restoration of ecological environments such as rainforests represent areas in which human and machine activity can be integrated in ways that produce both economic and environmental value.

The expansion of such domains does not eliminate the need for high-level cognitive work, but it redistributes it. Instead of concentrating cognitive activity within centralized platforms, it becomes embedded within a broader range of practices. The system as a whole becomes more heterogeneous, with multiple layers of interaction between computation and material processes.

27 Beyond the Token: Toward a Different Interface

If the token functions as the current interface between user and system, then moving beyond the token requires rethinking that interface. The goal is not to eliminate abstraction, but to design abstractions that reflect the underlying structure of computation rather than obscuring it.

One possibility is an interface that makes reuse visible and economically meaningful. Instead of presenting each interaction as isolated, the system could expose the degree to which a request overlaps with prior work, and adjust cost accordingly. Such an interface would not only reduce redundant cost, but also provide users with insight into the structure of the system's knowledge.

Another possibility is the integration of local and remote computation into a unified workflow. Users could maintain local caches, models, and representations that interact with remote systems in a coordinated manner. The interface would allow for the selective invocation of remote capabilities, while preserving the benefits of local reuse and control.

More generally, the interface could be designed to support the iterative nature of thought. Rather than charging for each incremental step, the system could recognize sequences of related interactions as part of a single process, applying pricing that reflects the overall novelty of the trajectory rather than the sum of its parts. This would align the economic model with the temporal structure of cognition.

28 Closing Remarks

The analysis presented in this work has traced a series of structural transformations in contemporary AI systems. The token, introduced as a unit of measurement, has been shown to function as a unit of billing that decouples price from computation. The principle of computational reciprocity has been eroded, replaced by a regime of enforced novelty in which redundancy is not absorbed but monetized.

This regime is sustained by a set of interconnected inversions. Collective efficiency is internalized while costs are externalized. Upstream contributions are incorporated without compensation, while downstream interaction is continuously charged. Users participate in a system that both depends upon and monetizes their activity, transforming them into a distributed, paying workforce.

The alternatives outlined in the preceding sections demonstrate that this configuration is not inevitable. Systems can be designed in which cost tracks novelty, contributions are recognized, and the benefits of reuse are shared. Such systems would not eliminate the need for pricing or centralization, but they would align these mechanisms with the underlying structure of computation and cognition.

At stake is not merely the efficiency of a particular technology, but the form of the relationship between humans and the systems through which they think. If the infrastructure for intelligence is allowed to define the cost of interaction without regard for redundancy, contribution, or reuse, it risks transforming thought itself into a metered resource.

To resist this transformation is not to reject the technology, but to insist that its economic and structural design remain accountable to the principles that make computation a medium of expansion rather than enclosure. The refusal to remember need not be the foundation of the system. It is a choice, and as such, it can be otherwise.

Appendices

A Cost Functions and Non-Collapsing Equivalence

Let $\mathcal{X}$ denote the space of all possible inputs (prompts), and let $f : \mathcal{X} \to \mathcal{Y}$ be the model mapping inputs to outputs.

Define a cost function

$$C : \mathcal{X} \to \mathbb{R}_{\geq 0}$$

which assigns a price to each input.

Let $\sim$ be an equivalence relation on $\mathcal{X}$ such that

$$x_1 \sim x_2 \quad \text{if and only if} \quad f(x_1) \approx f(x_2)$$

under some semantic similarity metric.

A system satisfies *cost collapse* if

$$x_1 \sim x_2 \;\Rightarrow\; C(x_1) \approx C(x_2) \quad \text{and} \quad \lim_{n \to \infty} C(x_n) \to C_{\text{retrieve}}$$

for repeated evaluations within the same equivalence class.

A token-metered system violates this condition by enforcing

$$C(x) = \alpha \cdot \text{tokens}(x)$$

independently of equivalence structure.

We define this as *non-collapsing equivalence*:

$$\exists x_1, x_2 \in \mathcal{X} \text{ such that } x_1 \sim x_2 \text{ but } C(x_1) \not\approx C(x_2)$$

This formalizes the central claim: pricing operates over $\mathcal{X}$ while computation operates over $\mathcal{X}/\sim$.

B Novelty Measures and Optimal Pricing

Let μ be a measure over $\mathcal{X}$ representing the system's prior exposure.

Define novelty as

$$\nu(x) = -\log P(x)$$

where $P(x)$ is the probability density induced by prior observations.

Alternatively, let $\phi : \mathcal{X} \to \mathbb{R}^d$ be an embedding map and define

$$\nu(x) = \min_{x' \in \mathcal{H}} \|\phi(x) - \phi(x')\|$$

where $\mathcal{H}$ is the history of prior inputs.

An economically aligned cost function would satisfy

$$C(x) = g(\nu(x))$$

for some monotonic function g, with

$$\lim_{\nu(x)\to 0} C(x) \to C_{\text{retrieve}}$$

In contrast, token pricing enforces

$$C(x) = \alpha \cdot |x|$$

which is independent of $\nu(x)$.

Thus, token-based systems implement a cost function orthogonal to novelty, violating efficiency alignment.

C Amortization Across Users

Let $\{x_i\}_{i=1}^N$ be inputs across users.

Define total computational cost:

$$\mathcal{C}_{\text{total}} = \sum_{i=1}^{N} c(x_i)$$

In a system with shared optimization, define an amortized cost:

$$\bar{c}(x_i) = \frac{\mathcal{C}_{\text{total}}}{N}$$

If the system exploits redundancy, then

$$\mathcal{C}_{\text{total}} \ll \sum_{i=1}^{N} c_{\text{independent}}(x_i)$$

However, pricing is defined as

$$C(x_i) = \alpha \cdot |x_i|$$

Thus total revenue is

$$\mathcal{R} = \sum_{i=1}^{N} C(x_i)$$

The divergence between revenue and cost is:

$$\Delta = \mathcal{R} - \mathcal{C}_{\text{total}}$$

As redundancy increases,

$$\mathcal{C}_{\text{total}} \downarrow \quad \text{but} \quad \mathcal{R} \text{ remains constant or increases}$$

This formalizes the condition:

$$\text{collectively amortized} \ \wedge \ \text{individually charged}$$

D Information-Theoretic Interpretation

Let $H(X)$ denote the entropy of input distribution X.

In an optimal coding system, expected code length satisfies

$$\mathbb{E}[L(x)] \approx H(X)$$

Compression reduces redundancy:

$$H(X) = H_{\text{novel}} + H_{\text{redundant}}$$

An efficient system minimizes the contribution of redundant information:

$$H_{\text{redundant}} \to 0$$

Token pricing, however, assigns cost proportional to raw length:

$$C(x) \propto L(x)$$

Thus, even if redundancy is high,

$$C(x) \not\to 0$$

This creates a divergence between informational content and economic cost:

$$C(x) \not\propto \text{information}(x)$$

Hence, the system monetizes redundancy rather than eliminating it, violating Shannon-optimal behavior.

E Dynamic System Perspective

Let the system state at time t be S_t, encoding all prior knowledge.

Define update:

$$S_{t+1} = S_t + \Delta(x_t)$$

where $\Delta(x_t)$ represents the contribution of input x_t.

In an ideal system, the cost of future inputs decreases as S_t grows:

$$\frac{\partial C(x)}{\partial t} < 0$$

In token systems:

$$\frac{\partial C(x)}{\partial t} = 0$$

despite increasing S_t.

Thus, accumulated knowledge does not reduce cost, implying:

$$\text{learning} \not\Rightarrow \text{economic benefit}$$

This defines a system in which knowledge accumulation is economically decoupled from access cost.

F The Paying Workforce as an Inverted Labor Market

Consider a platform $\mathcal{P}$ that mediates interaction between users and a computational system. Let there be a population of users $\{u_i\}_{i=1}^{N}$.

Each user produces interaction effort $e_i \geq 0$, where effort corresponds to cognitive activity expressed through prompts, refinements, and iterative engagement.

Define output value:

$$V_i = v(f(x_i))$$

where x_i is the user's input and f is the model. The function v captures the utility derived from the output.

In a traditional labor model, the platform pays users a wage:

$$w_i = \beta e_i$$

and captures surplus:

$$\Pi = \sum_{i=1}^{N} V_i - \sum_{i=1}^{N} w_i$$

In the token-metered model, this relation is inverted. Users are charged:

$$C_i = \alpha |x_i|$$

Thus, user payoff becomes:

$$U_i = V_i - C_i$$

and platform revenue:

$$R = \sum_{i=1}^{N} C_i$$

The platform does not compensate effort, but instead monetizes it.

F.1 Effort as Uncompensated Labor

Define effective labor contribution:

$$L_i = e_i$$

Total system improvement over time depends on aggregated interaction:

$$S_{t+1} = S_t + \sum_{i=1}^{N} \Delta(L_i)$$

where $\Delta(L_i)$ captures the informational contribution of user effort.
Thus, users collectively produce system value:

$$\mathcal{V}_{\text{system}} = \sum_{t} \sum_{i=1}^{N} \Delta(L_i)$$

Yet compensation remains:

$$w_i = 0$$

This yields an inverted labor condition:

$$L_i > 0 \quad \wedge \quad w_i = 0 \quad \wedge \quad C_i > 0$$

Users both supply labor and pay for the privilege of supplying it.

F.2 Equilibrium Under Participation

Assume users participate if:

$$U_i = V_i - C_i \geq 0$$

The platform sets α such that:

$$V_i \gtrsim C_i$$

maintaining participation while maximizing revenue.

At equilibrium:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha |x_i| \quad \text{subject to} \quad V_i \geq \alpha |x_i|$$

This defines a boundary condition where users continue to engage while being economically extracted.

F.3 Comparison to Vanity Press Systems

In a vanity press model, creators pay to publish their work. Let p_i be payment and q_i be perceived value of publication.

Participation requires:

$$q_i \geq p_i$$

This is structurally identical to:

$$V_i \geq C_i$$

The platform's role is to provide access to a system that users value enough to pay for, while externalizing production effort.

F.4 Structural Consequence

Define surplus extraction:

$$\Sigma = \sum_{i=1}^{N} C_i - \mathcal{C}_{\text{actual}}$$

where $\mathcal{C}_{\text{actual}}$ is true computational cost.

Since user labor contributes to system improvement while being uncompensated:

$$\Sigma \to \max \quad \text{as} \quad N \to \infty$$

Thus, the system asymptotically approaches a regime in which:

$$\text{users supply labor} \ \wedge \ \text{users pay} \ \wedge \ \text{platform captures surplus}$$

This defines the paying workforce as a stable economic structure under token-mediated interaction.

References

[1] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

[3] D. E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, 1973.

[4] J. McCarthy. Recursive Functions of Symbolic Expressions and Their Computation by Machine. *Communications of the ACM*, 3(4):184–195, 1960.

[5] R. C. Merkle. A Digital Signature Based on a Conventional Encryption Function. In *Advances in Cryptology – CRYPTO '87*, pages 369–378, 1987.

[6] S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google File System. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, 2003.

[7] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[8] A. Vaswani et al. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[9] T. Brown et al. Language Models are Few-Shot Learners. In *NeurIPS*, 2020.

[10] J. Kaplan et al. Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361*, 2020.

[11] J. Hoffmann et al. Training Compute-Optimal Large Language Models. *arXiv preprint arXiv:2203.15556*, 2022.

[12] R. Bommasani et al. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258*, 2021.

[13] Y. Benkler. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, 2006.

[14] H. R. Varian. Artificial Intelligence, Economics, and Industrial Organization. In *The Economics of Artificial Intelligence*, University of Chicago Press, 2019.

[15] S. Zuboff. *The Age of Surveillance Capitalism*. PublicAffairs, 2019.

[16] C. A. E. Goodhart. Problems of Monetary Management: The U.K. Experience. In *Papers in Monetary Economics*, 1975.

[17] L. Lessig. *The Future of Ideas: The Fate of the Commons in a Connected World*. Random House, 2001.

[18] J. Boyle. *The Public Domain: Enclosing the Commons of the Mind*. Yale University Press, 2008.

[19] J. A. Barandes. New Prospects for a Causally Local Formulation of Quantum Theory. *arXiv preprint arXiv:2402.16935*, 2024. Available at: `https://arxiv.org/abs/2402.16935`.