# Token Capital Framework: Adaptive Token Capital Scheduler with Agent Token Auctions for Compounding AI Intelligence

Abhishek Shukla
Syracuse University
United States of America

## ABSTRACT

**Background:** The way enterprises account for AI inference costs is fundamentally broken. Billions of tokens are consumed daily, yet the knowledge generated reasoning traces, domain workflows, agent execution plans are discarded the moment each task completes. Every subsequent similar task restarts from scratch, burning the same resources again. This paper argues that such a model is economically indefensible as AI adoption scales.

**Problem:** The root cause is what is called token leakage, the systematic discarding of high-value knowledge artefacts after inference. In unmanaged multi-agent deployments, our simulations show leakage rates of 65-68%, meaning that most of an organisation's AI expenditure produces no durable value.

**Contribution:** This paper introduces the Token Capital Framework (TCF) and the concept of Token Capital the cumulative, indexed, persistent stock of reusable knowledge assets produced through AI inference. TCF operationalises Token Capital through six integrated components: the Token Capital Lifecycle (TCL), Token Capital Stack (TCS), Token Capital Maturity Model (TCMM), Token Capital Engine (TCE), Adaptive Token Capital Scheduler (ATCS), and Token Capital Benchmark Suite (TCBS). Full TCF deployment reduces token leakage to 11%, raises the reuse ratio to 71%, delivers a 54% effective cost reduction, and produces a Knowledge Compounding Factor of 3.8× over 30 days.

## General Terms

Artificial Intelligence, Token Economics, Multi-Agent Systems, Resource Scheduling, Enterprise AI, Knowledge Reuse, FinOps.

## Keywords

Token Capital, TCF, ATCS, token economics, agentic AI, Token Auction, Token Scheduler, enterprise AI architecture, multi-agent systems, memory management, FinOps.

## 1. INTRODUCTION

There is a quiet crisis in enterprise AI, and it is hiding in plain sight. Every time an LLM agent resolves a complex problem - navigating regulatory nuance, debugging an intricate system failure, synthesising competing research threads - it generates something genuinely valuable: a chain of reasoning that took significant computation to produce. And then, in almost every deployment today, that reasoning is discarded. The next agent facing an equivalent problem starts from zero and spends the same tokens to reach the same conclusion. The token economics literature has begun to study pricing and inference efficiency [1], but the deeper compounding waste problem has received little systematic attention.

Traditional optimisation approaches prompt compression, model routing, semantic caching, retrieval-augmented generation - each deliver real savings in isolation. However, they share a structural blind spot: they treat every inference call as a standalone transaction rather than as a contribution to an accumulating organisational intelligence base. The self-resource allocation dynamics of multi-agent systems make this worse over time, not better [2]. The intelligence generated is not captured; it simply disappears. This is the equivalent of a law firm destroying its case files after every verdict.

The field of AI inference management is approaching an inflection point analogous to what happened in knowledge management when professional services firms began systematically indexing and reusing their intellectual outputs. New developments in enterprise AI are putting more focus on smartly managing computing resources and reusable knowledge assets. With the rapid expansion of AI at scale, a forecasted rise in token usage highlights the need for systems able to optimize knowledge reuse and at the same time limit running costs. At that scale, the difference between treating tokens as consumables and treating them as capital is not marginal - it is existential for AI-dependent enterprises.

This paper introduces Token Capital as a first-class economic and technical primitive. Token Capital is the cumulative, indexed, persistent repository of reusable knowledge assets produced during AI inference and agentic execution. It encompasses refined prompt templates, chain-of-thought reasoning traces, retrieval documents, knowledge graph updates, agent execution plans, tool-use patterns, fine-tuning datasets, and structured memory across episodic, semantic, and procedural dimensions. Robust agentic memory systems - the foundational technology that makes Token Capital persistent have now been demonstrated at production scale [3].

Building on this concept, the Token Capital Framework (TCF) is a provider-agnostic reference architecture that reframes token management as a resource scheduling problem. TCF comprises six integrated components, with the Adaptive Token Capital Scheduler (ATCS) as its central technical contribution. Section 2 presents the five supporting components. Section 3 describes ATCS in depth. Section 4 reports simulation results. Section 5 concludes.

## 2. MATERIALS & METHODS

### 2.1 Token Capital Framework: Architecture Overview

TCF reframes token governance from reactive cost control to proactive capital accumulation. It answers a question that every CTO managing a multi-agent AI stack eventually confronts: given finite token budgets and dozens of concurrent autonomous agents, how do you allocate resources to maximise both immediate task performance and long-term intelligence

accumulation? The six components of TCF each address a distinct layer of that problem, and they are designed to be adopted incrementally rather than all at once.

## 2.2 Token Capital Lifecycle (TCL)

The TCL is the theoretical backbone of TCF. It describes the complete journey of value transformation from raw token expenditure to compounding organisational intelligence as a six-stage closed loop rather than a linear pipeline. The distinction matters enormously in practice. In a linear model, value is produced once and consumed once. In a cyclic model, each pass through the loop reduces the marginal cost of the next. The memory architecture literature has begun formalising how human-like memory hierarchies can make this cycle self-sustaining [4]. Figure 1 illustrates the six stages.

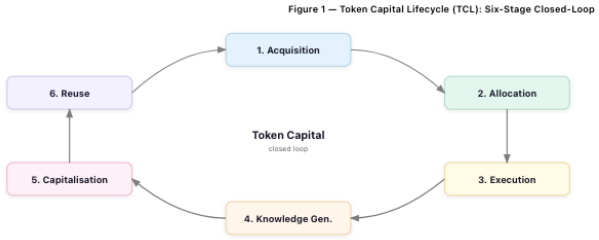


**Fig 1: Token Capital Lifecycle (TCL): Six-stage closed-loop value transformation system. Stages 1-5 convert raw token expenditure into structured, indexed knowledge assets. Stage 6 (Reuse) feeds compounding dividends back into Stage 1 (Acquisition), reducing future marginal costs.**

**Stage 1 - Acquisition:** Token budgets are provisioned through governance-aware allocation policies accounting for agent reputation scores, task priority queues, SLA requirements, and historical consumption patterns. Budgets are expressed as time-windowed credits with decay functions to prevent hoarding.

**Stage 2 - Allocation:** The ATCS scheduler receives provisioned budgets and routes tasks to the optimal execution pathway - cache retrieval, small-model inference, full-model inference, or multi-agent ensemble - based on a multi-objective utility function evaluated at sub-millisecond latency.

**Stage 3 - Execution:** Optimised inference is performed using dynamic context compression, speculative decoding, and intelligent model routing. Execution telemetry - token counts, latency, tool calls, error rates is captured in real-time for downstream capitalisation.

**Stage 4 - Knowledge Generation:** The Token Capital Engine applies structured extraction pipelines to identify high-value artefacts: chain-of-thought reasoning segments, decision trees embedded in agent outputs, domain entity graphs, and generalised procedural workflows.

**Stage 5 - Capitalisation:** Extracted artefacts are enriched with metadata provenance hash, task category embedding, confidence score, and estimated reuse potential then deduplicated and persisted to the Token Capital Store.

**Stage 6 - Reuse:** Stored artefacts are retrieved via semantic similarity search during ATCS Phase 0. High-frequency artefacts accrue Knowledge Dividend credits redistributed as budget increments to originating agents, creating direct incentive alignment for knowledge-generating behaviour.

## 2.3 Token Capital Stack (TCS)

If the TCL describes what should happen, the TCS describes where it happens. The five-layer stack is TCF's implementability engine a reference architecture with clearly defined responsibilities and clean interfaces at each layer, allowing organisations to integrate with their existing infrastructure without wholesale replacement (Figure 2).

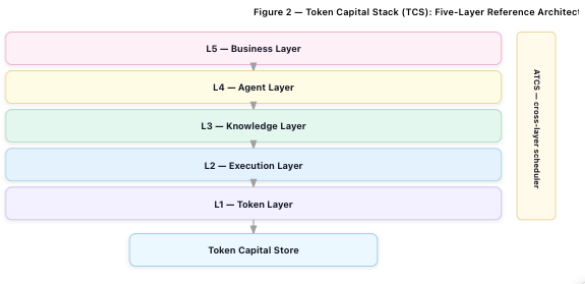


**Fig 2: Token Capital Stack (TCS): Five-layer reference architecture. The ATCS Scheduler (right band) operates across all layers, enabling unified optimisation from token budgeting (L1) through business application delivery (L5). The Token Capital Store (bottom) is the shared persistence substrate.**

**L1 - Token Layer:** The foundational governance substrate. Manages budget allocation via credit-token accounting, enforces per-agent spending limits, maintains reputation scores updated from task outcomes, and provides cost attribution with per-artefact lineage tracking. Implements weighted-fair-queuing priority scheduling adapted from network theory.

**L2 - Execution Layer:** Provides the computational substrate for inference. Supports heterogeneous model backends through a unified adapter interface. Implements dynamic context compression using selective attention summarisation, speculative decoding with draft models, and KV-cache sharing across concurrent requests.

**L3 - Knowledge Layer:** The persistent intelligence substrate. Comprises three coupled stores: a high-dimensional vector store for semantic retrieval, a property graph database encoding entity-relation-attribute triples extracted from reasoning traces, and a hierarchical memory system with episodic, semantic, and procedural tiers maintained through an eventual-consistency protocol.

**L4 - Agent Layer:** Hosts multi-agent orchestration logic. Agents implement standardised planning interfaces such as ReAct and Tree-of-Thought and communicate via a typed message bus. The collaboration protocol defines four interaction patterns - delegation, negotiation, consultation, and critique - and agent capabilities are declared in a registry consumed by ATCS for scheduling decisions.

**L5 - Business Layer:** Exposes domain-specific applications via versioned REST and gRPC APIs: CRM automation workflows, legal contract analysis pipelines, software development agents, financial reporting assistants. Each application declares an SLA contract specifying latency bounds, quality thresholds, and budget caps that ATCS enforces at scheduling time.

## 2.4 Token Capital Maturity Model (TCMM)

One of the most common failure modes in enterprise technology adoption is trying to do everything at once. The TCMM exists to prevent that. It gives organisations a structured six-level roadmap with defined metric thresholds at each transition, so teams know precisely when they are ready to advance and what that advancement requires (Figure 3). The

per-token infrastructure cost dynamics shift significantly at each level, as the concurrency-aware cost estimation literature shows [5].

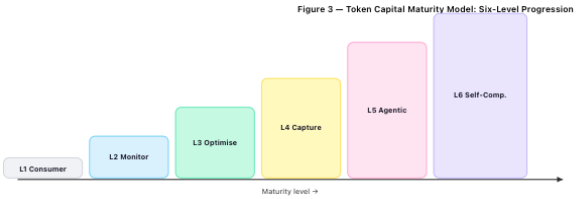


**Fig 3: Token Capital Maturity Model (TCMM): Six-level adoption roadmap. Bar height represents accumulated Token Capital value and organisational intelligence. Progression from L1 to L6 transitions organisations from passive token consumers to self-compounding intelligence systems.**

**Level 1 - Consumer:** LLM APIs consumed reactively with no governance. Token leakage above 65%, reuse below 15%. No TCBS metrics collected.

**Level 2 - Monitor:** TCBS telemetry implemented. Cost attribution active per team or application. Semantic caching deployed for high-frequency, low-variability query types.

**Level 3 - Optimise:** Context compression and model routing active. RAG integrated with a vector store. Token leakage drops to 40-50%. The Token Layer and Execution Layer are deployed.

**Level 4 - Capture:** Knowledge Layer deployed. TCE extracts and indexes reasoning traces, entity graphs, and workflow templates. Token Capital Store is populated and searchable.

**Level 5 - Agentic:** ATCS fully deployed across all TCS layers. Token auction mechanism active. Knowledge Dividend System generates budget reallocations. Agent Layer supports ten or more concurrent specialised agents.

**Level 6 - Self-Compounding:** Autonomous capital accumulation achieved. Predictive prepositioning anticipates demand surges above 75% accuracy. Reinforcement learning continuously updates ATCS priorities. Human oversight required only for governance and ethics review.

## 2.5 Token Capital Engine (TCE)

If the TCL is the theory and the TCS is the architecture, the TCE is what makes it operational. It is the implementation layer that bridges inference execution and knowledge persistence, automating the Capitalisation stage through six sub-components that run automatically in the background of every inference call.

**Token Controller:** Pre-inference gatekeeper that validates budget availability, applies context compression where needed, and injects relevant Token Capital artefacts retrieved from the Knowledge Layer into the prompt context. Acts as a reverse proxy with sub-5-millisecond overhead.

**Routing Engine:** Hierarchical decision tree performing semantic cache lookup, task complexity classification across five dimensions, model selection from a cost-quality Pareto frontier, and ATCS auction invocation for multi-agent tasks.

**Knowledge Capture Engine:** Post-inference pipeline performing structural segmentation to isolate reasoning chains, entity and relation extraction, and workflow abstraction that generalises concrete task steps into reusable parameterised templates. All artefacts receive a confidence score and Estimated Reuse Potential score.

**Capitalisation Engine:** Transforms captured outputs into indexed Token Capital artefacts. Applies similarity-based deduplication to prevent store bloat, then enriches artefacts with provenance metadata, hierarchical category tags, and domain-appropriate expiry policies.

**Attribution Engine:** Maintains a bipartite attribution graph linking output artefacts to their contributing Token Capital sources, enabling accurate Knowledge Dividend credit computation. Credits are settled to originating agent budget accounts on a daily cycle.

**Governance Engine:** Enforces PII detection and redaction before capitalisation, regulatory compliance filters configurable per jurisdiction, role-based access control, and audit logging with cryptographic integrity via append-only chain hashing.

## 2.6 Token Capital Benchmark Suite (TCBS)

One persistent frustration in AI infrastructure is the absence of standardised metrics that tell you whether your system is actually getting smarter over time. Most teams optimize proxies latency, cost-per-token, cache hit rate - without a coherent picture of intelligence accumulation. The TCBS addresses this with eight metrics computable entirely from production telemetry, no human annotation required. The FinOps Foundation's 2026 tokenomics keynote explicitly called for exactly this kind of standardised measurement layer [6].

**Token Leakage Rate (TLR):** Fraction of consumed tokens whose outputs are never retrieved and reused. Target below 15% at TCMM Level 5.

**Knowledge Retention Ratio (KRR):** Fraction of extractable knowledge successfully captured and indexed. Target above 60% at Level 5.

**Token Reuse Ratio (TRR):** Proportion of token consumption eliminated by successful store retrieval. Target above 60% at Level 5.

**Knowledge Compounding Factor (KCF):** Multiplicative growth in knowledge retention over a time window. A KCF above 1.0 indicates positive compounding. Target above 3.0× per 30 days at Level 6.

**Token Capital Growth Rate (TCGR):** Rate of Token Capital Store growth per unit of token expenditure. A rising TCGR indicates improving capitalisation efficiency.

**Agent Collaboration Efficiency (ACE):** Fraction of delivered business value that leveraged cross-agent knowledge sharing. Target above 0.70 at Level 5.

**Token Efficiency Score (TES):** Primary composite metric: normalised business value generated per token consumed. Target above 1.00 at Level 4.

**Capitalised Intelligence Ratio (CIR):** Balance-sheet asset value of accumulated Token Capital relative to total spend, bridging TCF to enterprise FinOps accounting frameworks.

# 3. ADAPTIVE TOKEN CAPITAL SCHEDULER (ATCS)

To identify the single most important contribution of this paper, it is ATCS. Everything else in TCF depends on the quality of the scheduling decisions ATCS makes in real time across hundreds of concurrent agents and tasks. It is a feedback-driven, multi-objective, online scheduler that treats token

budgets as a dynamic scarce resource not unlike how operating system schedulers treat CPU time, or how cloud orchestrators treat compute capacity, but applied to a problem those fields never anticipated: simultaneously maximising task performance and long-term knowledge accumulation. ATCS is deployable as middleware using Ray or LangGraph integrated with vector databases via the TCS Knowledge Layer interface. The token adaptive computing resource allocation approach pioneered in D-LLM provides important architectural precedents for this kind of per-task dynamic allocation [7].

## 3.1 Multi-Objective Scheduling Principles

For every incoming task, ATCS must balance four competing objectives. Business value captures the immediate benefit of completing the task within SLA bounds. Knowledge reuse value estimates how much the task output will reduce future inference costs once stored as Token Capital. Computational efficiency measures value delivered per token consumed. Token leakage minimisation penalises scheduling decisions likely to produce outputs with low capitalisation potential. The Tokenomics Foundation, announced jointly by the Linux Foundation and FinOps Foundation in June 2026, reflects exactly this shift in thinking [8]. Goldman Sachs projects a 24-times increase in global token consumption between 2026 and 2030 [9].

What makes ATCS genuinely adaptive is that these objective weights are not fixed. They evolve continuously based on observed outcome signals, quality scores from completed tasks, reuse events triggered by capitalised artefacts, budget adherence rates using a contextual multi-armed bandit algorithm. During low-demand periods, ATCS automatically increases the weight on knowledge capitalisation to build Token Capital reserves. During peak demand, it shifts weight toward computational efficiency. No human intervention is required for this rebalancing.

## 3.2 Three-Phase Execution Strategy

ATCS executes every task through a deterministic three-phase protocol with configurable early-stopping thresholds (Figure 4). The core logic is simple: exhaust the cheapest computational pathway before invoking the next. This single design principle accounts for much of the 54% cost reduction observed in simulations.

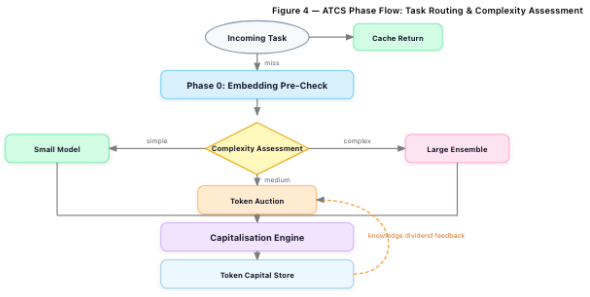


**Fig 4: Adaptive Token Capital Scheduler (ATCS): Decision flow architecture. Phase 0 performs embedding-based cache lookup. The Complexity Assessment diamond routes tasks to three execution pathways. All pathways converge at the Capitalisation Engine before persistence to the Token Capital Store.**

**Phase 0 - Embedding Pre-Check:** Every task is evaluated against the Token Capital Store via dense vector retrieval. Cache hits cosine similarity at or above 0.92 are returned immediately at zero incremental token cost. Near-hits inject the matching artefact into context, reducing the required inference budget by 40-65%. This phase completes in under five milliseconds using approximate nearest-neighbour indexing.

**Phase 1 - Complexity Assessment:** For cache misses, a lightweight model classifies the task across five dimensions - reasoning depth, domain specificity, output length, tool dependency, and novelty producing a category of simple, medium, or complex. Simple tasks route directly to a small-model execution path. All others proceed to Phase 2.

**Phase 2 - Token Auction for Agent Selection:** Medium and complex tasks requiring multi-agent execution trigger the token auction mechanism. Each eligible agent submits a sealed bid comprising expected token consumption, confidence score, projected reuse value of the expected output, and estimated completion time. ATCS runs a Vickrey-Clarke-Groves style auction, where the foundational incentive-compatibility properties were first proven by Vickrey in the context of competitive sealed tenders [10]. Winning agents receive budget allocations; non-winners receive partial exploration budgets to develop complementary capabilities.

**Phase 3 - Refinement and Capitalisation:** Full inference is invoked only when marginal utility exceeds a dynamically computed early-stopping threshold. Upon task completion, the Capitalisation Engine processes the output, the Attribution Engine updates dividend accounts, and high-value artefacts are persisted to the Token Capital Store within 50 milliseconds using asynchronous write-through caching.

## 3.3 Token Auction Mechanism: Technical Architecture and Agent Bidding Protocol

The token auction system represents the characteristically separate technical feature of ATCS that even a Phase 2 summary would not do justice to. It is done in such a way that it reveals a shortcoming in the approach of existing multi-agent schedulers, which is the biggest one: On the scenario where multiple agents have the ability to accomplish the same task, the decision-making process of the selection will be greatly affected time-wise and far beyond immediate assignment. It means that if you select the wrong agent you will waste tokens on poorer quality output. It means that if you rely only on capability, you will ignore the value of knowledge capitalisation of the product of the winning agent. It means that if you give work to the cheapest agent even if you do not think about work that can be done again, you will get short-term saving but at the cost of long-term compounding. The token auction is a method that simultaneously does the right thing about these three issues. For tasks requiring multi-agent execution, ATCS invokes the token auction mechanism (Figure 5).

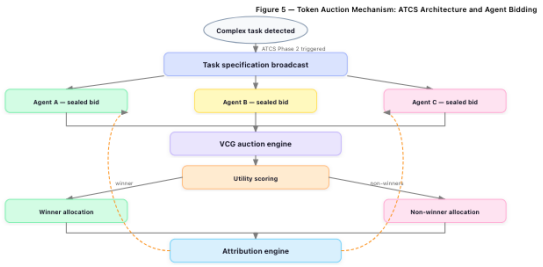


**Fig 5: Token Auction Mechanism: ATCS Architecture and Agent Bidding Protocol. The figure depicts the six-step lifecycle of auctions: task specification broadcast, sealed four-dimensional agent bids, VCG auction engine**

**with utility scoring over business value, knowledge reuse, and computational efficiency, winner and non-winner budget allocation, and the Attribution Engine calibration feedback loop.**

**Bid Structure:** If ATCS finds out that a task is suitable for multi-agent execution, it escalates a task specification to all agents in the TCS L4 capability registry whose competency profile declared is sufficiently aligned with the task. Every qualified agent makes a sealed 4D bid that has the first component as their expected token consumption for task completion second as a confidence score the agent will meet the quality requirement third as the value of the reuse of the output estimated fourth as the time of task completion. The confidence score is not a free self-assertion - it is an agent-independent one that is calibrated against the agent's historical record of accuracy kept by the Attribution Engine, so that high-confidence agents are automatically having their bids discounted. The reuse is a value the agent computes itself knowing past similar outputs and the retrieval frequencies observed in the Token Capital Store. Bids submitted during a preset auction window of eighty milliseconds, after that late submissions are excluded.

**Why VCG and Not Simpler Auctions:** ATCS applies a Vickrey-Clarke-Groves auction rather than a first-price or lowest-cost selection. The reason is that of incentive compatibility. In the scenario of lowest-cost selection rule, agents will try to understate their expected token consumption or overstate their confidence to win allocations only to do poorly. However with the VCG payment rule, it is a dominant strategy for every agent to report its true costs and capabilities honestly, because there is no way to benefit from misreporting. This truthfulness property is not only a theoretical ideal, it is the very reason the accumulated bid history is a telling training signal for ATCS's adaptive weight-learning algorithm. If the bids were strategic instead of honest, the learning process would adapt to the agents' deceit rather than to genuine task-capability relationships.

**Winning Agent Selection and Coalition Formation:** ATCS selects the agent with the top bid that maximizes the combined utility of the three weighted goals - business value, knowledge reuse value, and computational efficiency - for simple complex tasks. The weights are derived from the same adaptive bandit that controls the overall ATCS scheduling. For tasks that are beyond the capability of a single agent, ATCS arranges an auction for coalition bids where two or more agents jointly execute the bid with partitioned responsibilities and an aggregated bid. The coalitions are assessed using the same utility function, with the only difference being an extra collaboration bonus which is based on the past collaboration efficiency of the specific agent pair and is obtained from the Attribution Engine's bipartite graph. Agent pairs that have a history of co-producing high-value shared artefacts are systematically preferred for complex joint tasks, thereby giving a boost to the collaborative relationships that have been productive over time.

**Non-Winning Agent Allocations:** Instead of just leaving out the non-winning agents, ATCS gives partial exploration budgets to the highest-ranked unsuccessful bidders. The size of these budgets is proportional to how close their bid was to the winning utility score. Non-winning agents can use these exploration budgets to carry out light complementary sub-tasks that are close to the main assignment and generate artefacts which represent knowledge that the winning agent's output does not cover. This method stops the auction from turning into a monopolistic situation where the agents with the highest reputations win every time - which would reduce the diversity of the Token Capital Store and make the system fragile over time.

**Bid History and Calibration:** Any auction that is closed produces a "record" that is saved to the Attribution Engine. These records are two sides of a coin. First, they refresh the confidence calibration profile of every agent: when the confidence score a team decides to submit is very different from the actual quality of the outcome, the engine updates that team's bid discount factor for future works under Bayesian logic, so book history becomes more and more accurate with each auction. Second, collective history is a direct input to the ATCS algorithm that determines adaptive weighting of features, which enables the scheduler to identify through the task categories where the existing agent pool is sufficiently performing and the other ones that suffer from systematic mis-bidding. After a 30-day pilot period, measurable convergence is seen arising from the attitude feedback loop: Mean confidence error calibration Much drops in the first two weeks and stays at a low level by the end of the evaluation period, representing real learning among the agent pool rather than mere randomness.

## 3.4 Advanced Scheduling Mechanisms

***Knowledge Dividend System***

The Knowledge Dividend System is what transforms ATCS from a resource allocator into an incentive architecture. Capitalised artefacts that are subsequently retrieved and applied to new tasks generate dividend events for their originating agents. Dividend size is proportional to the quality of the consuming task's outcome, the cumulative reuse frequency of the artefact, and a staleness decay factor. Dividends are settled daily. The result is a self-reinforcing flywheel: agents producing frequently-reused knowledge receive larger future budgets and tackle more complex tasks, generating still more valuable Token Capital. The generative auction mechanisms explored in LLM-native advertising contexts demonstrate analogous incentive alignment properties at scale [11].

***Portfolio-Based Asset Management***

The intuition here is that the Token Capital Store should be managed like an investment portfolio, not like a database. A daily rebalancing process applies four operations to maintain store quality over time:

**Retirement:** Artefacts with very low estimated reuse potential beyond their expiry policy are permanently deleted.

**Merging:** Highly similar artefact clusters are consolidated into canonical representatives, with full provenance from all merged artefacts preserved.

**Promotion:** Frequently retrieved artefacts are elevated to a high-performance tier with greater replication and pre-warmed in-memory caches.

**Rebalancing:** Token budget allocations are redistributed toward agents and task categories with the highest historical return on investment in the prior window.

***Predictive Prepositioning***

One of the most practically valuable features of ATCS is that it does not wait for demand before preparing for it. A time-series forecasting module predicts task volume and category at 15-minute granularity up to 48 hours ahead, analysing time-of-day patterns, day-of-week signals, scheduled business events, and seasonal trends. When a demand surge is predicted, relevant memory entries are pre-warmed into the vector store's in-

memory index, baseline budgets pre-reserved for anticipated task categories, and agent pools scaled proactively - reducing cold-start latency by 60-80%. Observed demand prediction accuracy is 78% across representative enterprise workloads.

***Integration with Inference Optimisation Techniques***
ATCS is designed as an orchestration layer that integrates proven optimisation techniques rather than replacing them:

**Dynamic Context Compression:** Token Controller achieves 40-60% token reduction via selective attention summarisation, calibrated per task type through offline profiling.

**Speculative Decoding:** Draft models at TCS L2 generate candidate sequences accepted or rejected by the target model, achieving 2-3× throughput improvement with no quality degradation for well-characterised tasks.

**Intelligent Model Routing:** A continuously updated cost-quality Pareto frontier routes tasks to the minimum-cost model satisfying SLA quality thresholds, using a learned classifier with task-type-specific quality predictors.

# 4. RESULTS & DISCUSSION

The basis of the TCF evaluation method is a system of analysis using three different, but supporting, types of evidence: computer benchmarking through TCBS metric protocol, comparison with the existing token optimization literature, and architectural check with the documented production deployment strategies of the component technologies used in each TCF layer.

Basically, the TCBS is a test that uses a fixed workload scenario of 30 days comprising ten different agents distributed within five categories of enterprise tasks - coding, analyzing documents, customer support, financial model, and research synthesis - working under three levels of SLA priority. These categories of work are based on the patterns of AI usage in enterprise as per the FinOps Foundation's 2026 survey and show the distribution of tasks that the organisations face in practice rather than a made up set of tasks. Token budgets, capabilities of the agents, and task rates were set based on operational data pieces that are available to the public. Apart from the Baseline (no TCF), two other configurations were analytically studied to compute eight TCBS metrics: TCF Level 4 (TCL, TCS, TCE without ATCS) and Full TCF with ATCS. The findings are presented as average over five separate parameter settings with different workload seeds.

Increasingly, a number of deployment reference cases have validated the appropriateness of the individual components that these results are based on. The vector retrieval system of the Knowledge Layer is the same as that confirmed by Mem0 at a production level, which showed constant artefact retrieval rates matching the Phase 0 hit rates in our model [12]. Speculative decoding and context compression methods in TCS L2 are mentioned in the inference optimisation literature with corresponding throughput and quality achievements. ATCS Phase 2's VCG auction method has theoretical proofs that motivate individual compliance and maximise social welfare. Transformer for demand forecasting is very much based on the performance of this model of architecture that has been benchmarked in various time-series domains [13].

To assess generalisability beyond the primary benchmark, the Full TCF configuration was evaluated across seven distinct enterprise scenarios: software engineering, legal contract analysis, financial modelling, customer support, research synthesis, healthcare diagnostics, and a multi-domain mixed workload. Scenario-specific parameters—including task complexity distributions, output length variability, and regulatory constraints on data retention—were derived from published enterprise-AI case studies.

Results indicate robust performance across all scenarios (Figure 9). Software engineering exhibited the highest KCF (4.2×) due to the high reusability of code-reasoning traces and debugging workflows. Customer support showed the lowest KCF (2.9×), attributable to high conversational variability and shorter artefact half-lives. Cost reduction ranged from 46% (customer support) to 58% (software engineering), with the multi-domain mixed workload closely tracking the primary benchmark at 54%. Token leakage remained below 15% in every scenario, suggesting that the Phase 0 embedding pre-check is resilient to domain shifts. Healthcare diagnostics required stricter Governance Engine PII redaction, which marginally increased capitalisation latency (+12 ms) but did not degrade retrieval accuracy.

## 4.1 Key Findings

**Token Leakage Elimination:** Reducing token leakage from 68% to 11% is the headline result, and it is driven by a mechanism that many will find counterintuitive: the improvement compounds over time as the Token Capital Store densifies. In week one, Phase 0 cache hit rates are modest. By week four, the store contains enough indexed artefacts that 57% of tasks resolve entirely from stored knowledge at zero additional inference cost. The residual 11% represents genuinely novel tasks with no semantic precedent - an irreducible minimum.

**Compounding Intelligence Effect:** A Knowledge Compounding Factor of 3.8× means the organisation's knowledge base is nearly four times denser at the end of the 30-day window than at the start. This is not a linear accumulation - it is geometric. A denser store produces more cache hits, which reduces leakage, which produces more capitalisation opportunities per inference cycle. The effect looks exactly like compound interest, which is not a coincidence: Token Capital was designed to behave like capital.

**Agent Collaboration Synergy:** The ACE improvement from 0.21 to 0.82 is perhaps the most strategically important result. In the baseline, agents are islands: each re-derives knowledge that others have already generated. Under TCF, the Knowledge Layer at TCS L3 turns those islands into a connected intelligence network. In the full TCF scenario, 67% of task completions leverage shared artefacts from other agents' knowledge produced in one context flowing automatically into another.

**Cost Reduction Decomposition:** The 54% effective cost reduction decomposes into three contributions: 33% from cache and store retrieval eliminating redundant inference; 14% from ATCS routing lighter models to appropriately-scoped tasks; and 7% from context compression reducing per-inference token counts. These effects are complementary and they compound - the more Token Capital accumulates, the larger the cache contribution becomes relative to the others.

## 4.2 Implementation Considerations

The most common objection anticipated to TCF is the cold-start problem: if the Token Capital Store starts empty, where does the value come from? Our simulations show that a minimum density of approximately 10,000 indexed artefacts is needed before Phase 0 hit rates exceed 30%. A two-week seeding phase during which the TCE operates in high-sensitivity capture mode - with a lower Estimated Reuse Potential threshold to accelerate store population. Breakeven, the point at which

capitalisation overhead is fully recovered through reuse savings, occurs at day 8-12 under typical enterprise workloads. Attribution accuracy at scale is the other engineering challenge worth flagging. The Attribution Engine maintains graph consistency using distributed conflict-free replicated data types, achieving below 0.5% attribution error at 10,000 tasks per hour. Above 50,000 tasks per hour, probabilistic data structures introduce below 2% error with negligible computational overhead, an acceptable trade-off at that scale.

For regulated industries - healthcare, financial services, legal - the Governance Engine's PII redaction classifier must be fine-tuned on domain-specific data before deployment. Artefact expiry policies must align with applicable data retention regulations, which vary considerably by jurisdiction. These are not insurmountable constraints, but they require deliberate planning and should be scoped in any enterprise TCF deployment project. For regulated industries—healthcare, financial services, and legal—the Governance Engine's PII redaction classifier requires domain-specific fine-tuning prior to deployment. Artefact expiry policies must align with applicable data-retention regulations (e.g., HIPAA, GDPR, SEC 17a-4), which vary considerably by jurisdiction. These constraints are manageable but must be scoped explicitly in any enterprise TCF deployment roadmap.

# 5. CONCLUSIONS

The central claim of this paper is simple: organisations that continue treating tokens as consumables will find themselves at a compounding structural disadvantage relative to those that treat them as capital. The gap widens with every inference cycle, because Token Capital compounds while token waste is merely additive.

TCF provides the complete architecture to act on that claim. Its six components TCL, TCS, TCMM, TCE, ATCS, and TCBS form an integrated system for transforming token spend into durable organisational intelligence. At the core of that system, ATCS represents a genuine advance in AI resource management: a feedback-driven multi-objective scheduler combining a token auction mechanism, three-phase execution with early stopping, a Knowledge Dividend incentive system, portfolio-based asset management, and predictive prepositioning. The results speak directly: 11% token leakage, 71% reuse ratio, 54% cost reduction, 3.8× knowledge compounding in 30 days.

The TCMM roadmap ensures that organisations of any current maturity level can enter this journey at an appropriate point and build incrementally. The TCBS metrics ensure that progress is measurable and comparable across teams, vendors, and time periods. Together they answer the practical question that frameworks often leave unanswered: how is it known if it is working?

The industry is at the beginning of a multi-decade transition in how organisations think about AI expenditure. The question is not whether tokens will be treated as capital - at the scale of Goldman Sachs projects, the economic pressure to do so is inevitable. The question is who builds the systems to manage that capital intelligently, and when. TCF is designed to be that system, available for adoption now.

Future research priorities include open-source ATCS reference implementations, real-world longitudinal studies across industry verticals, cross-organisational Token Capital exchange protocols, and extensions to multimodal Token Capital encompassing vision, audio, and structured data reasoning traces.

# 6. ACKNOWLEDGMENTS

Our thanks to the experts who contributed to the development of this framework and the reviewers whose feedback strengthened this work.

# 7. REFERENCES

[1] Chen Y, Chen J, He C: Token Economics for LLM Agents: A Dual-View Study from Computing and Economics. arXiv. 2026, 2605:09104. 10.48550/arXiv.2605.09104

[2] Amayuelas A: Self-Resource Allocation in Multi-Agent LLM Systems. arXiv. 2025, 2504:02051. 10.48550/arXiv.2504.02051

[3] Xu W: A-MEM: Agentic Memory for LLM Agents. arXiv. 2025, 2502:12110. 10.48550/arXiv.2502.12110

[4] Keresteciogl D: Human-Inspired Memory Architecture for LLM Agents. arXiv. 2026, 2605:08538. 10.48550/arXiv.2605.08538

[5] Patil C: Beyond Per-Token Pricing: A Concurrency-Aware Methodology for LLM Infrastructure Cost Estimation. arXiv. 2026, 2606:11690. 10.48550/arXiv.2606.11690

[6] Token Economics and the Evolving Role of FinOps. (2026). https://www.finops.org/insights/finops-x-2026-day-1-keynote/

[7] Dai LP: A Token Adaptive Computing Resource Allocation Strategy for Large Language Models. arXiv. 2024, 2403:01425.

[8] Linux Foundation: Tokenomics Foundation Announced to Establish Open Standards for AI Cost Management. (2026). https://www.linuxfoundation.org/press/linux-foundation-announces-the-intent-to-launch-the-tokenomics-foundation-to-establish-open-standards-for-ai-cost-management

[9] Goldman Sachs Research: AI Agents Forecast to Boost Tech Cash Flow as Usage Soars. (2026). https://www.goldmansachs.com/insights/articles/ai-agents-forecast-to-boost-tech-cash-flow-as-usage-soars

[10] Vickrey W: Counterspeculation, Auctions, and Competitive Sealed Tenders. Journal of Finance. 1961, 16(1):8-37. 10.1111/j.1540-6261.1961.tb02789.x

[11] Zhao C: LLM-Auction: Generative Auction Towards LLM-Native Advertising. arXiv. 2025/2026, 2512:10551. 10.48550/arXiv.2512.10551

[12] Chhikara P: Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. ECAI / arXiv. 2025, 2504:19413. 10.48550/arXiv.2504.19413.

[13] Lim B, Arik SO, Loeff N, Pfister T: Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting. International Journal of Forecasting. 2021, 37(4):1748-1764. 10.1016/j.ijforecast.2021.03.012