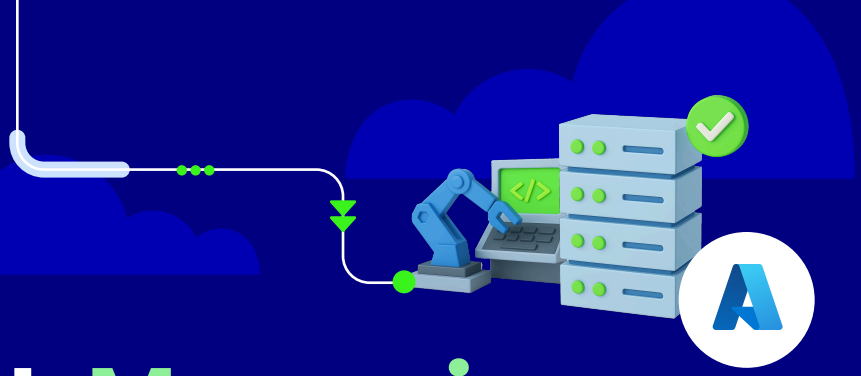
PointFive



# FinOps for AI: Managing LLM Costs in Azure OpenAI

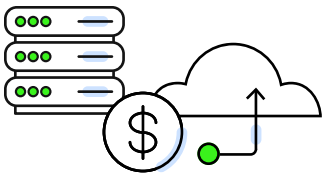
AI adoption is accelerating across every industry. All of the big three cloud providers make this possible with fully managed advanced models, deployed natively within the existing organizational cloud infrastructure. Using these services, companies can create new value for customers and employees by embedding large language models (LLMs) into products and workflows. Alongside the technology's rapid adoption are rapid cost increases. Industry surveys show AI budgets rising more than 30% year-over-year. In many organizations, LLM spending already represents the fastest-growing category of cloud infrastructure.

In this white paper, we'll explore how to tackle cost management and optimization for Azure OpenAI, a popular cloud-provider managed LLMs service. Although most concepts are common across cloud providers, Azure OpenAI has unique properties that pose significant challenges for FinOps practitioners trying to control, manage, and optimize AI spend.

Azure OpenAI's billing structures are complex; this is part of the challenge. Azure OpenAI's billing model and cost reporting tools lack the clarity companies need to lower costs through improved efficiency. Without strong cost controls, spend can scale faster than revenue.

Traditional FinOps practices cannot address these growing challenges. Traditional FinOps was designed for more predictable workloads like compute and storage, not the volatile, variable nature of AI. The industry now needs a new framework to understand the economics of Azure LLM workloads.

This whitepaper explores that framework. It outlines the challenges of Azure's pricing model, the business impact of unmanaged spend, and the market dynamics shaping AI cost management. It explains why use case economics matter more than token prices and how FinOps is evolving for GenAI.



# The Core Challenge

## Managed LLMs Pricing

Understanding the economics of managed LLM services is crucial for effective cost management. Unlike traditional cloud resources, LLM pricing is granular and can vary based on several factors:

## Tokens

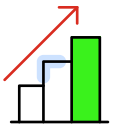
Every interaction with a large language model consumes tokens, and each token has a price. "Tokens" are the fundamental units of text that a model processes.

- **Input Tokens:** These are the tokens in the prompts, instructions, and contextual information that you send to the LLM. You are charged for every token consumed by the model as part of your input.
- **Output Tokens:** These are the tokens that the LLM generates as a response. You are charged for each token produced by the model.
- **Cached Input Tokens:** Some AI platforms support prompt caching within a session or conversation, meaning previously processed prompt prefixes (especially system messages, context, or long static instructions) are reused for subsequent turns. This reduces input-token costs because the model doesn't re-process those parts fully each time.

## Deployment Locality

Deployment Locality refers to the level of geographic control you choose for where data is processed and stored, balancing performance, compliance, and cost.

- **Data Zone Deployment (US or EU zones):** This option sits midway between Global and Regional, designed for scenarios that need both higher performance and locality compliance. Processing occurs across multiple regions within your chosen zone (for example, all EU or all US regions), providing a balance of compliance and performance. While it comes at a slightly higher cost than Global, it offers stronger compliance controls and still delivers better throughput than Regional.
- **Regional Deployment:** Regional deployment is the strictest locality option. It ensures that both processing and storage occur within a single Azure region, such as Australia or Germany. Pricing is generally comparable to the data zone option, but performance is lower due to the tighter regional constraints.



### Provisioned Throughput

Provisioned throughput is a capacity-based model in Azure OpenAI where you reserve dedicated resources for your LLM workloads. Instead of paying only for tokens used, you purchase Provisioned Throughput Units (PTUs), which guarantee a fixed level of input and output token capacity per second. This model offers guaranteed capacity, lower latency, and predictable costs, making it well suited for production workloads with consistent and scalable demand.

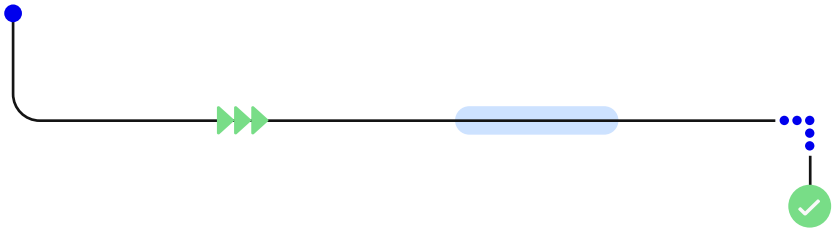
While it provides performance benefits, provisioned throughput typically comes at a premium compared to on-demand pricing. You pay for the reserved capacity, regardless of actual utilization. If your usage fluctuates significantly, underutilized provisioned throughput can lead to unnecessary costs. Conversely, exceeding your provisioned capacity may result in overflow traffic being billed at on-demand rates, or throttling - depending on configurations.

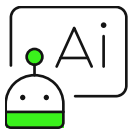
Understanding these pricing components is the first step in effectively managing LLM costs and preventing unexpected expenses.

## Lack of Granular Cost Visibility

Native Azure cost data is too high-level to support unit economics. In Azure OpenAI, two resource types exist, accounts and deployments. Accounts are administrative units, similar to clusters. Deployments represent the single model endpoint that applications call to run prompts, such as a GPT-4 or GPT-35-Turbo instance with a defined configuration. Configuration and usage live at the deployment level. Costs, however, aggregate to the account level. This obscures the link between costs and cost drivers.

If multiple applications share one Azure OpenAI Account, even best practices like separating model deployments fail to solve the problem of cost visibility. Native tools still cannot attribute costs by application. Without this level of visibility, optimization, model modernization, and capacity planning remain reactive instead of proactive. The result is bills that look complete but provide no guidance on where to cut or how to improve efficiency.






# Evolving FinOps for GenAI

It's clear that LLM workloads demand a new approach to cost management. Closing these gaps requires a virtual cost layer. This is a mapping layer that reveals the true drivers of spend by linking technical metrics to cost. It connects usage patterns and configuration choices to financial outcomes, making it possible to measure efficiency in real-time. Without this layer, optimization remains reactive, and teams only discover inefficiency after the bill arrives.

The virtual cost must later include accurate measurement beyond token totals, encompassing every factor that shapes the request cost. This includes the compounding cost of context windows, the premium charged for provisioned capacity, the differences between on-demand and reserved pricing, and the use case itself.

In addition to the virtual cost layer, tracking cost per business outcome, such as resolving a customer query or generating a design draft, will provide a common frame of reference for finance and engineering. This outcome-based metric links the technical work of tuning prompts and selecting models with the financial goals of controlling budgets and protecting margins.

FinOps for GenAI requires continuous alignment. Engineering decisions about architecture, latency, and accuracy now carry direct financial consequences. Finance teams must understand the technical drivers behind spend, while engineers must see the financial impact of their choices. Without this shared perspective, optimization remains reactive and budgets remain unstable.

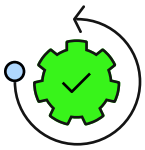
# The Business Impact

LLM workloads scale with customer adoption. Each new feature release or product enhancement increases usage, which drives infrastructure costs higher. While this growth is the goal, many organizations see increased expenses before an increase in revenue. This presents a challenge. The majority of that growth comes from compute and storage tied directly to model usage, confirming that LLM costs are now one of the fastest growing categories of cloud spend.

**Margins erode when costs grow faster than revenue. Three blind spots make this worse:**

- Unclear optimization priorities. Teams do not know which use cases to focus on.
- Limited understanding of usage behavior. Spikes, idle periods, and unpredictable demand make capacity decisions risky.
- Opaque cost drivers. Native Azure tools obscure the link between deployment and application.

Making data-driven decisions regarding AI features' financial viability requires data. Only once this data is provided can optimization become possible.



# Optimizations Techniques

LLM cost optimizations can be categorized into several key areas:

- **Rate** is the baseline pricing you pay for model usage. This can be reduced by leveraging commitment discounts (e.g., reservations and savings plans), similar to traditional cloud practices.
- **Infrastructure configurations** are the deployment parameters that determine how your Azure OpenAI resources are set up. These settings, including rate limiting and provisioning size, directly influence cost and efficiency without changing the model's behavior or outputs.
- **Model selection** is the choice of model type and version. This decision directly affects performance, cost, and token pricing. In some cases, selecting a newer model within the same tier and capability set, such as moving from o1 to o3, can provide savings without reducing response quality.
- **Model interaction** is the way applications engage with the model through prompts that include context and instructions that wrap end user input. Designing system prompts and context formats with cost in mind can reduce input and output tokens without compromising results.

# Beyond Pricing: Why Use Case Economics Matter More

Token usage is the most visible driver of Azure LLM costs, but it is also highly variable. But tokens are only part of the equation. The total cost of ownership extends beyond token usage. A single application may combine multiple AI models, supporting cloud services, and integration layers. Each of these components adds cost, and together they define the true economics of a use case. Within this broader context, small technical adjustments create outsized impact. Prompt tuning can shorten inputs, model swaps can lower token costs, and architectural changes can shift how often an LLM is called. Each change alters not just token spend but also the way surrounding systems contribute to total cost.

Unit economics must therefore be anchored in the use case. A business outcome, such as resolving a support ticket or generating a code suggestion, provides a more reliable measure of efficiency than the raw price per token. Engineering and finance teams need to align around this outcome-based view to make decisions that balance accuracy, latency, and cost.

# Cloud Efficiency Posture Management: The new approach

Managing LLM economics requires more than extending existing FinOps practices. It demands a framework that integrates financial data with technical usage metrics and provides real-time visibility into efficiency. This is the role of Cloud Efficiency Posture Management, or CEPM.

Though built with more traditional FinOps strategies in mind, CEPM is well suited to optimize Azure OpenAI. CEPM is built on three core principles:

- **Visibility at the right level.** CEPM breaks spend out of aggregate account views and exposes the unit economics of individual workloads.
- **Continuous alignment.** CEPM creates a shared language for finance and engineering, connecting technical design choices to financial outcomes.
- **Proactive optimization.** CEPM shifts teams from reacting to monthly bills to simulating tradeoffs and planning changes before costs escalate.

PointFive is the pioneer of CEPM platforms. By using the PointFive platform, organizations can seamlessly embed CEPM into their daily workflows. CEPM brings great benefit to organizations utilizing Azure OpenAI.

### Granular Cost Visibility in AI Deployments

Since native Azure billing reports spend at the account level, they hide the activity of individual deployments.  PointFive surfaces costs at the deployed model level, information hidden in Azure billing, using a virtual cost layer.

The virtual cost layer can fit perfectly in PointFive's  Data Fabric paradigm, powering CEPM. The Data Fabric correlates billing exports, configurations, and operational metrics and implements novel transformations to produce an accurate virtual cost layer. This virtual cost layer is then used both as a foundation for potential savings of cost optimization opportunities, and for analytics, exposing metrics such as effective cost per request.

With this virtual cost layer, teams can attribute spend to specific applications, track unit economics, and measure the value of provisioned throughput against simulated on-demand costs. Instead of seeing one large, blended bill, organizations can finally see which apps and models are driving costs and adjust their usage with confidence.

## Model Selection and Modernization

PointFive scans environments to identify outdated models, compares their usage to the cost and capabilities of newer ones, and simulates the financial impact of switching. In many cases, replacing an older model with a modern version can deliver both better performance and significant savings (replacing o1 with o3 for up to 80–90% savings). PointFive makes it easy to see when you are paying more for less, so you can upgrade to faster, cheaper models without the guesswork.

## On-Demand vs. Provisioned Throughput

In production, on-demand services with different rates for input and output can fall short on throughput and latency, degrading end-user experience. Provisioned throughput often costs more per token because most workloads never hit full capacity. It carries a premium for guaranteed high availability, yet utilization rarely justifies the spend. Teams either end up paying extra for performance they don't fully use or risk slower service when demand spikes.

CEPM helps teams simulate the tradeoffs between cost, utilization, and performance by comparing actual workload patterns against the economics of provisioned units. It also factors in reservation discounts, which can reduce costs by more than 60 percent when applied to stable workloads. By modeling both modes side by side, PointFive's CEPM platform ensures that throughput decisions are based on economics as well as performance.

## Making the Most of Your Provisioned Throughput

### Right-Sizing Deployments

Provisioned throughput should match real workload patterns, not theoretical peak demand. Metrics from Azure Monitor show how much of the provisioned capacity is actually used. PointFive adds another layer by mapping these utilization patterns to cost, making it clear when a deployment is oversized. By comparing observed usage against allocated units, teams can scale deployments to the right size and reduce waste.

### Eliminating Non-Production Waste

Development, test, and QA environments rarely need guaranteed throughput. Running them on provisioned capacity drives costs up without delivering value. PointFive helps identify these environments by linking deployment metadata with billing data. Moving non-production workloads to on-demand services cuts premium spend immediately while preserving performance in production.

# Conclusion

True efficiency comes from understanding how tokens, throughput, and supporting cloud services combine into the full economics of a use case. But this information isn't included in native Azure billing. Without this view, costs rise faster than revenue and margins erode.

Traditional FinOps practices, while effective for compute and storage, fall short when applied to LLM workloads. The volatility of token usage, the opacity of account-level billing, and the complexity of provisioned capacity demand a new approach.

CEPM provides that approach. CEPM connects usage metrics to financial outcomes, exposes hidden inefficiencies, and aligns engineering and finance around shared measures of efficiency. It turns opaque billing into actionable insight.

Organizations that adopt CEPM can scale AI features with confidence, knowing that throughput decisions, model choices, and architecture changes are evaluated against both cost and performance. This posture allows teams to protect margins while still delivering the responsiveness and accuracy their customers expect.

## PointFive is pioneering CEPM tools.

Learn more about our features at www.pointfive.co or by emailing us at **hello@pointfive.co**

PointFive