Microsoft Foundry Quota Management as an Agent Ops Control Plane

Foundry quotas (TPM/PTU) let you allocate capacity across deployments and projects. That allocation layer is a monetizable governance surface for enterprise agent operations.

Source https://yetyield.com/blog/microsoft-foundry-quota-management-as-an-agent-ops-control-plane/

Enterprises don’t buy “more tokens.”

They buy controlled capacity:
• which teams get how much throughput
• which deployments are allowed to be production
• who can request increases
• how quota changes propagate and get audited

Microsoft Foundry makes this sellable because quota is treated as an allocatable resource across deployments.

The monetization angle

Sell a “Foundry quota governance” offer:
• Implementation sprint: map workflows to deployments, allocate quota, define escalation rules, and document operational responsibilities.
• Monthly retainer: rebalance quota as usage shifts, handle throttling, and manage quota increase requests as planned events.

This extends the operations cluster:
• How to Price Agent Platform Operations Retainers (Without Hand-Wavy AI ROI) (/blog/how-to-price-agent-platform-operations-retainers/)
• How to Sell Agent Spend Controls and Stop-Loss Rules as an Ops Retainer (/blog/how-to-sell-agent-spend-controls-and-stop-loss-rules-as-an-ops-retainer/)

What Foundry documents (officially)

In Microsoft Foundry (new), quota is:
• assigned per subscription, per region, and per model, in tokens per minute (TPM)
• managed differently across deployment types (for example, Standard vs Provisioned)
• editable in the Foundry portal via a Quota view, with a separate view for Provisioned throughput unit (PTU) allocations

The docs also describe:

• a shared quota pool for short-term testing (not for production endpoints)
• quota propagation delays (up to ~15 minutes)
• the roles required to view and request quota changes

Official reference: https://learn.microsoft.com/en-ie/azure/foundry/how-to/quota?view=foundry

Why quota allocation is an enterprise wedge

In early-stage products, “who gets to use the model” is informal.

In enterprise, it is governance:
• quota is a budget proxy
• quota is a risk control
• quota is a political allocation problem

That’s why quota management creates recurring work.

A quota-first blueprint for agent deployments

Step 1: classify deployments by intent

Create a clear taxonomy:
• Experiment: shared quota or low-cap standard deployment
• Staging: limited quota, used for regression tests and rollout gates
• Production: dedicated quota, approvals required for changes

Foundry documentation explicitly notes shared quota should be used for testing inferencing and not production endpoints.

Step 2: allocate quota by workflow criticality

Quota allocation is your pricing wedge:
• revenue/compliance workflows get reserved capacity and higher ceilings
• low-stakes workflows get shared or capped quota

This mirrors the retainer pricing axes (criticality, tool risk, change frequency).

Step 3: design an escalation ladder

Quota requests are operational events:
• who can request increases (Owner/Contributor roles)
• what evidence is required (usage graphs, forecast, incident history)

• what approvals are needed (finance/security)

The “quota request packet” is a buyer-facing artifact you can sell.

Turning quota into a retainer

Quota doesn’t stay stable because:
• new teams adopt the system
• prompts drift and token footprint changes
• model choice shifts
• incidents trigger temporary throttles or rebalancing

Monthly deliverables
• quota allocation report (what changed, why)
• usage hotspots (deployments approaching ceilings)
• throttling incidents summary + mitigations
• next month’s rebalancing plan

Optional upsell: controlled rollout playbooks

Bundle quota governance with:
• evaluation gates
• approval workflows
• incident response runbooks

Quota is how you enforce the boundary; governance is how you keep it enforceable.

What to do next

Foundry quota governance pairs well with a second layer: billing budgets and alerts.

If you want a cross-platform extension, add Google Cloud Billing budgets and Vertex AI quotas to build a “dual-layer” stop-loss package (quota + spend alerts).