

Estimating the cost of Fabric data agents: a worked method

Companion to 'Building Enterprise AI Agents on Microsoft Fabric: Architecture, Security and Production Lessons', Leon Gordon, Onyx Data.

Free at onyxdata.co.uk/fabcon, with no registration.

A free companion paper to the FabCon Europe 2026 session "Building Enterprise AI Agents on Microsoft Fabric: Architecture, Security and Production Lessons" (session TH14, 1 October 2026, Barcelona).

Leon Gordon, Onyx Data. Paper dated 26 September 2026. Prices are public retail list prices in GBP for the UK South region, taken from the Azure Retail Prices API on 15 September 2026. Product statements were checked against Microsoft Learn on the dates given beside each one and in the sources table.

1. Summary

A Fabric data agent that answers questions for business users can be paid for in two quite different ways, and most early estimates go wrong because they mix the two.

- **Route A, Fabric-managed.** A Copilot Studio agent calls the Fabric data agent as a tool. The data agent's model work runs inside Fabric and is charged as capacity units (CU) against a Fabric capacity you have already bought. The cost is a capacity, bought in steps, plus a few smaller lines.
- **Route B, customer model.** Your own agent (for example a Foundry agent) reasons with a model deployment you own, such as Azure OpenAI, and pays per token. The cost is tokens, plus hosting, plus telemetry, plus the Fabric capacity your data still needs.

These are alternatives. They are never added together.

This paper sets out one method for both routes. It starts from a demand chain (how many people, how often, how many retries, how many useful answers), turns that into the quantity each route actually bills for, prices it at dated public rates, and then tells you exactly which lines it cannot price. It ends with a fictional worked case, Northgate Mutual, whose figures are reproduced exactly by the free companion calculator and by a Python twin of the same engine.

For the fictional reference workload of 1,000 eligible users, the method gives these priced-line subtotals per month:

	Route A (Fabric-managed)	Route B (customer model)
What you buy	F128 capacity, one-year reservation, UK South	GPT-5.4 tokens, App Service hosting, trace ingestion

	Route A (Fabric-managed)	Route B (customer model)
Priced lines, GBP per month	8,593.21	3,569.11
Per useful answer	GBP 0.15	GBP 0.064
Lines left unpriced	Copilot Credits, source queries, licences	Fabric capacity for the data, hosted-agent compute, networking

Read the last row before the others. The two totals are not a like-for-like comparison until the unpriced lines have been measured, and an unknown line is not a zero.

2. Who this paper is for

It is written for the people who are asked "what will this cost?" before anything has been built: data platform leads, Fabric capacity administrators, solution architects, FinOps analysts and technical buyers. It assumes you know what a Fabric capacity and an F SKU are, and roughly what a token is. It does not assume you have run a data agent in production. The interactive calculator and this paper are free at onyxdata.co.uk/fabcon, with no registration.

It is not a Microsoft pricing guide, a quote, or a forecast of anyone's bill. It structures an estimate so that every number is a published rate, a visible assumption, or an explicit unknown.

3. The two charging routes (and why you never add them)

The first question in any estimate is: **who bills the inference?**

Route A: the model runs inside Fabric

On route A, a Copilot Studio agent (in Teams, for example) calls the Fabric data agent through the Fabric IQ Data MCP tool. The data agent uses a model to interpret the question and write queries. Microsoft Learn's data agent consumption page says this model work is metered in CU seconds per 1,000 tokens, and that "the AI related activity within the Data agent is classified as background jobs" (read 16 and 23 September 2026). That CU is drawn from your Fabric capacity. You do not receive a separate token bill for it; you pay for the capacity.

Route A's bill of materials:

1. **Fabric capacity** (the dominant line), bought in SKU steps, at pay-as-you-go or reservation rates.
2. **Copilot Credits** for the Copilot Studio agent, billed separately from the capacity unless a Microsoft 365 Copilot licence covers the use (see 5.6). The unit price is published; the quantity per request is not.
3. **Source-engine query CU** for the queries the data agent generates, drawn from the capacity that hosts each source item (the same capacity in the reference case; our assumption).
4. **OneLake storage**, priced per GB-month.
5. **Licences** for Microsoft 365, Teams and Copilot Studio, which depend on your tenant.

Route B: the model is yours

On route B, your own agent reasons with your own model deployment. In the reference case this is a Foundry agent on a GPT-5.4 deployment, fronted by a web app. You pay per token on the deployment's meters.

Route B's bill of materials:

1. **Model tokens** (the dominant line): input, cached input and output, per million.

2. **Hosting** for the web app (App Service in the reference case).
3. **Trace ingestion** for the telemetry you keep.
4. **A Fabric capacity for the data.** The semantic model or lakehouse still lives on a capacity, sized by the source query load rather than by AI Query CU. The reference case does not price this line.
5. **Hosted-agent compute, evaluation runs, networking and private endpoints**, which the reference case does not price.

Why the totals are never added

Each route prices the same questions. Adding them would pay twice for one answer. The companion calculator shows both routes side by side, against the same denominator, and never shows a combined total.

There is one legitimate case where lines from both routes appear in a single estimate: when a route B agent calls the Fabric data agent as a tool. Then each request has two inference steps on two meters: tokens for your model, and AI Query CU for the data agent's model on a Fabric capacity. The calculator handles this with a switch that adds the capacity line to route B once. Note that the Foundry Fabric data agent tool is listed as preview on Microsoft Learn (read 23 September 2026).

4. The demand chain

Both routes start from the same six numbers. Every one of them is an assumption until you replace it with a measurement.

Step	Reference value	What to replace it with
Eligible users	100 or 1,000	the approved rollout population
Daily active share	60%	sign-in or first-question telemetry
Questions per active user per workday	5	transcript counts
Workdays per month	22	your business calendar
Retried questions	10% (each retry a full priced attempt)	attempts divided by questions, from your request log
Useful answers	85% of questions	your request log: the answer matched a reference and was accepted

From these:

- **Questions per month** = eligible users × daily active share × questions per active user per day × workdays.
- **Priced attempts** = questions × (1 + retried share). Every retry is billed in full, on either route.
- **Useful answers** = questions × useful-answer share.
- **Cost per useful answer** = all priced lines of one route ÷ useful answers.

The numerator carries every attempt, including retries and failures. The denominator carries only useful outcomes. That asymmetry is deliberate: when answer quality falls, the cost per useful answer rises, which is what a budget holder actually experiences. A useful answer here means one the user accepted and that matched an independently held reference, not merely a request that completed without error.

For 1,000 eligible users: $1,000 \times 0.60 \times 5 \times 22 = 66,000$ questions; $\times 1.10 = 72,600$ priced attempts; $\times 0.85 = 56,100$ useful answers. Note that 1,000 users is tenfold demand, not a thousand people asking at the same moment. Concurrency is a separate question (section 5.7).

The last input is the **request shape**: how many tokens a typical attempt reads and writes. The reference shape is 20,000 input tokens and 1,000 output tokens per attempt; a lighter shape is 5,000 in and 500 out. Both are assumptions. Large schemas, long instructions and conversation history push input tokens up quickly, which is why the shape is usually the most sensitive single input.

5. Route A method: Fabric capacity

5.1 CU seconds per attempt, at the published rates

Microsoft Learn's data agent consumption page (updated 22 July 2026; read 16, 23 and 25 September 2026) publishes one flat rate per token class for the data agent's "AI Query" operation:

Token class	CU seconds per 1,000 tokens
Input prompt	100
Cached input prompt	10
Output completion	400

So, for one attempt:

$$\text{CU seconds} = (\text{uncached input tokens} \times 100 + \text{cached input tokens} \times 10 + \text{output tokens} \times 400) \div 1,000$$

For the reference shape with no caching: $(20,000 \times 100 + 1,000 \times 400) \div 1,000 = \mathbf{2,400 \text{ CU seconds per attempt}}$.

For the light shape: $(5,000 \times 100 + 500 \times 400) \div 1,000 = \mathbf{700 \text{ CU seconds}}$.

Three cautions:

- The page itself says "Consumption rates are subject to change at any time." Treat this as a method, and re-read the page before you rely on a figure.
- Per-model token rates do exist in Fabric, but for different operations: AI Functions and Foundry Tools in Fabric publish per-model CU rates (read 16 September 2026; the Foundry Tools in Fabric page is itself marked preview). Do not transfer those rates to a data agent estimate. As of 23 September 2026, the data agent page publishes the flat 100, 10 and 400.
- The calculator's default cached share is 0%, because we have no measured cache rate to use. A 50% cached share is shown as a sensitivity.

5.2 The Fabric denominator is a day, not a month

Monthly CU hours = attempts $\times$ CU seconds per attempt $\div$ 3,600. For the reference case: $72,600 \times 2,400 \div 3,600 = 48,400$ CU hours a month, or 2,200 CU hours per workday.

The workday figure is the one that matters. Learn's throttling and smoothing page (updated 18 August 2026; read 18 September 2026) says Fabric "smooths background operations over a 24-hour period", in 30-second timepoints, 2,880 of them in a day. Because the data agent's AI work is a background operation (data agent consumption page; Fabric operations page, which lists "AI query" under the LlmPlugin item as Background on the Copilot and AI meter, read 16 September 2026), a busy workday's AI demand has to fit inside the capacity's next 24 hours.

A capacity's day is its CU count $\times$ 24 CU hours. Learn's own example is that an F2 provides 2 CU seconds per second, "meaning 48CU hours per day" (surge protection page, read 16 September 2026). The calculator applies the same rule up the F SKU ladder, so an F64 day is 1,536 CU hours and an F128 day is 3,072.

The consequence, which is our reading of the 24-hour smoothing rule rather than a Microsoft sentence, is simple and unforgiving: **weekend and paused hours cannot be borrowed**. A capacity that is idle on Saturday does not bank CU for Monday. The monthly total is a useful volume check; it is the wrong basis for choosing a SKU.

5.3 The planning ceiling

The **planning ceiling** is the share of a SKU's 24-hour window you are willing to let the agent's background AI work use, leaving the rest for interactive source queries and every other workload on the capacity. The reference case uses 75%. That figure is an assumption, not guidance; set it from your own headroom policy.

5.4 SKU choice

The smallest SKU is the first one where $\text{SKU day} \times \text{planning ceiling} \geq \text{AI CU hours per workday}$.

SKU	CU hours per day	Reference shape (2,200 per workday)	Light shape (641.67)
F16	384	573%	167%
F32	768	286%	84%
F64	1,536	143%	42%
F128	3,072	72%	21%
F256	6,144	36%	10%

For the reference shape, an F64 cannot carry the load (143% of its day). F128 at a 75% ceiling allows 2,304 CU hours, which covers 2,200, and the AI work alone uses 72% of the F128 day **before a single source query runs**. The light shape fits an F64 at 42%.

The calculator's SKU ladder runs from F2 to F2048, with the SKU number equal to its CU count. Learn's Fabric licences page lists F2 to F8192 on the same basis (read 26 September 2026); the reference cases never go above F256.

5.5 Reservation versus pay-as-you-go

The calculator prices capacity in one of two ways, selectable on the page:

- **Pay-as-you-go:** $\text{SKU} \times \text{GBP } 0.154611 \text{ per CU hour} \times \text{billed hours}$. This is the retail price on the Compute Pool Capacity Usage CU meter in UK South; the Copilot and AI Capacity Usage CU meter lists the same price (Azure Retail Prices API, 15 September 2026). An F128 for 720 hours is GBP 14,248.95.
- **One-year reservation (the default):** $\text{SKU} \times \text{GBP } 805.448187 \text{ per CU per year} \div 12$. An F128 is GBP 8,591.45 a month.

The reservation rate comes from the Azure Retail Prices API item with productName "Fabric Capacity Reservation", meterName "Fabric Capacity CU" and reservationTerm "1 Year", UK South, GBP. The snapshot of 15 September 2026 returned 805.448187; the same query on 16 September 2026 returned 805.4482, which is the same figure at four decimal places. The API labels these items with a unit of measure of "1 Hour"; the per-CU-per-year reading is supported by arithmetic on the published figures (the three-year item is exactly three times the one-year item).

What Microsoft Learn's reservations page says (read 16 September 2026):

- Reservations are bought per Azure region, in whole CU increments, for one or three years, at a chosen scope.
- The discount "is applied to CUs on an hourly basis": unused reserved hours do not carry forward. If you reserve fewer CUs than you run, the remainder bills at pay-as-you-go.
- "A reservation doesn't cover storage or networking charges."

What it does not say: what happens to a reservation while the capacity is paused, and whether a reservation covers capacity overage. Neither is documented on the pages we checked, so model both, and do not assume either way. Capacity overage, where enabled, is its own meter in UK South at GBP 0.463832 per CU hour, three times the base rate, and Learn lists the feature as an opt-in preview (read 17 September 2026).

A useful rule of thumb, which is our arithmetic and not a Microsoft statement: $\text{GBP } 805.448187 \div (\text{GBP } 0.154611 \times 8,760 \text{ hours}) = 0.595$. On compute alone, a one-year reservation costs about what pay-as-you-go would cost if the capacity ran about 60% of the hours in a year. Below that uptime, with a verified pause schedule, pay-as-you-go tends to win; above it, reserving the CUs you run every hour and keeping bursts on pay-as-you-go tends to win. Remember that pausing does not make smoothed work disappear: Learn's pause and resume page says the remaining smoothed operations "are summed, and added to your Azure bill" when you pause.

5.6 Storage and the other route A lines

- **OneLake storage:** GBP 0.01767 per GB-month (OneLake Storage Hot Data Stored, UK South, Azure Retail Prices API, 15 September 2026). The reference case assumes 100 GB: GBP 1.77.
- **Copilot Credits:** GBP 0.007362 per credit (Pay As You Go Copilot Credit, UK South, Azure Retail Prices API, 15 September 2026). On Copilot Studio's GitHub Copilot harness, Learn says credits are charged "from the moment you start building", including previewing, testing and evaluations, and that this usage is not included in a Microsoft 365 Copilot licence. Standard-harness agents start billing after publish, and Learn lists their use in Microsoft 365 channels as covered for users with a Microsoft 365 Copilot licence (read 16 and 26 September 2026). Learn's billing overview gives estimated ranges by task complexity, not a per-request rate. The calculator therefore leaves this line **unknown** unless you enter a quantity.
- **Allocation view:** AI CU hours $\times$ GBP 0.154611 (GBP 7,483.17 a month in the reference case) is shown as the share of the capacity the AI work would use. It is never an extra line. Adding it to the capacity would count the same CU twice.

5.7 Concurrency is a separate question

The daily window answers "is there enough capacity over a day?". It does not answer "will users wait at 10 o'clock?". The calculator keeps a separate concurrency panel. With 3,300 attempts per workday over 8 active hours and a peak hour at four times the average, the peak hour carries 1,650 attempts; at 45 seconds end to end, about 21 are in flight at once. All four inputs are assumptions.

Learn's throttling page says interactive operations are smoothed over "a minimum of five minutes, and up to 64 minutes", far shorter than the 24-hour background window, and that under the default policy the first stage users feel is a 20-second delay on new interactive requests after 10 minutes of overage protection (read 18 September 2026). Our working assumption, for semantic model sources, is that the queries a data agent generates behave as interactive load, so they are the part most exposed to a burst. Other engines classify their operations differently; check the Fabric operations page for your source. Size that lane from a measured burst, read in Capacity Metrics, not from an average.

5.8 Route A unknowns

- Copilot Credits per request (unit price published; quantity not).
- Source-engine query CU on the capacity that hosts each source (depends on the generated queries; unmeasured).
- Microsoft 365, Teams and Copilot Studio licences (tenant-specific).
- Pause settlement of smoothed usage, reservation behaviour while paused, and overage.

A governance note: as of 23 September 2026 we found no AI-specific or per-agent spend cap for Fabric on the surge protection, capacity settings or Copilot and agent admin settings pages (a dated absence, not proof that none

exists). Learn documents capacity-level surge protection and workspace CU limits over a rolling 24 hours; it calls the workspace limits soft, because they are checked every five minutes. Copilot Studio, a different product and meter, documents an agent-level hard stop (read 16 September 2026).

6. Route B method: customer model

6.1 Tokens

Tokens per month = attempts × tokens per attempt, split into uncached input, cached input and output. Price each at the deployment's meters. The reference case uses the GPT-5.4 Global meters listed for UK South (Azure Retail Prices API, 15 September 2026):

Meter	GBP per 1M tokens
5.4 inp Gl 1M Tokens (input)	1.840604
5.4 cd inp Gl 1M Tokens (cached input)	0.18406
5.4 opt Gl 1M Tokens (output)	11.043622

For 1,000 users on the reference shape: 72,600 attempts × 20,000 = 1,452,000,000 input tokens and 72,600 × 1,000 = 72,600,000 output tokens. Tokens = 1,452.0 × 1.840604 + 72.6 × 11.043622 = **GBP 3,474.32**, or GBP 0.047856 per attempt.

Cached input is priced at one tenth of input on both routes (10 against 100 CU seconds in Fabric; 0.18406 against 1.840604 GBP on this meter). That ratio is our arithmetic on the published figures.

6.2 Hosting

App Service Premium v3 P1v3 (Linux), UK South, GBP 0.12737 per hour (Azure Retail Prices API, 15 September 2026). The reference case assumes one instance for 720 hours: GBP 91.71. Your hosting may differ entirely; this line exists so that it is not forgotten.

6.3 Trace ingestion

The reference case assumes 20 KB of trace data per attempt and prices it at the Analytics Logs Data Ingestion meter, GBP 2.120375 per GB (UK South, Azure Retail Prices API, 15 September 2026). For 72,600 attempts that is 1.452 GB, GBP 3.08. Two assumptions sit inside this line: that your telemetry resource bills against this meter, and the 20 KB figure. The API also lists a zero-priced tier for the first units; the calculator ignores it, which slightly overstates this small line.

6.4 What route B does not price

- **Fabric capacity for the data.** The customer model still queries data that lives on a Fabric capacity. That capacity is sized by the source query load, not by AI Query CU, and the reference case does not model it. This line alone can change the comparison.
- Foundry hosted-agent compute, evaluation runs, networking and private endpoints.
- Identity, support, tax and operations labour.

If your route B agent calls the Fabric data agent as a tool, switch the calculator's tool setting on. The capacity line is then added to route B once, because the data agent's own model work bills AI Query CU to a Fabric capacity. For the reference case that takes route B to GBP 12,160.56 a month, GBP 0.22 per useful answer.

7. Sensitivities

Change one input at a time and watch which lines move. For 1,000 users on the reference shape (route A at the one-year reservation rate):

Change	SKU (share of its day)	Route A priced, GBP	Route A per useful	Route B priced, GBP	Route B per useful
Base case	F128 (72%)	8,593.21	0.15	3,569.11	0.064
Daily active 40%	F128 (48%)	8,593.21	0.23	2,409.97	0.064
Daily active 80%	F256 (48%)	17,184.66	0.23	4,728.24	0.063
3 questions a day	F128 (43%)	8,593.21	0.26	2,178.15	0.065
8 questions a day	F256 (57%)	17,184.66	0.19	5,655.55	0.063
Light shape 5k / 500	F64 (42%)	4,297.49	0.077	1,163.81	0.021
Retries 0%	F128 (65%)	8,593.21	0.15	3,252.98	0.058
Retries 25%	F256 (41%)	17,184.66	0.31	4,043.30	0.072
Useful answers 70%	F128 (72%)	8,593.21	0.19	3,569.11	0.077
Useful answers 100%	F128 (72%)	8,593.21	0.13	3,569.11	0.054
Cached input 50%	F128 (45%)	8,593.21	0.15	2,366.46	0.042
Planning ceiling 60%	F256 (36%)	17,184.66	0.31	3,569.11	0.064

Four patterns are worth carrying away:

1. **Route A moves in steps; route B moves in proportion.** Halving demand on route A often changes nothing until you drop below a SKU boundary; a small increase can double the capacity line. Route B scales smoothly with attempts.
2. **Retries hurt twice on route A.** At 25% retries, attempts rise enough to push the reference workload over the F128 ceiling, doubling the capacity line, while useful answers stay the same.
3. **The useful-answer share moves both routes' cost per answer** without changing either bill. It is the cheapest lever to measure and the one most often left at an optimistic guess.
4. **Request shape dominates.** The light shape cuts route B tokens by about 70% and drops route A a SKU.

8. The calibration step

The worked figures are an illustration. They become an estimate only when replaced with your own measurements.

First, measure one request. Run a representative question through the route you are evaluating, on a capacity or deployment you control.

- **Route A:** read it in the Microsoft Fabric Capacity Metrics app. Learn's data agent consumption page says the app shows data agent usage "under the name AI Query", and the app's Timepoint item detail page includes filters for operation ID, user and CU thresholds (read 25 September 2026). Reading one request's AI Query line combines those two pages; no single sentence on Learn describes it. The Compute page reports CU seconds by item and operation over 14 days. Allow for latency: usage data generally appears 10 to 15 minutes after the activity, and a new capacity, workspace or item may not appear until the scheduled overnight refresh (read 18 September 2026). Copilot Credits are read separately, in the Power Platform admin

centre's Copilot Studio capacity management; Learn notes that discrete costs are not attributed to individual makers or end users.

- **Route B:** read it in Cost Management cost analysis, grouped by Meter. Learn's Foundry cost page says you "typically see meters that account for model input and output usage" when you group by Meter (read 25 September 2026). Cost Management shows these costs **by day, month or year, not per request**; Learn advises using trend windows rather than minute-by-minute comparisons. So calibrate route B from a day of known traffic, divided by that day's attempts, not from a single call. We have not performed this read in any tenant ourselves. If you are reconciling a Fabric reservation, note that Learn says the purchase appears in cost analysis as "Dataflows Standard Compute Capacity Usage CU", with charge type Purchase; that is a route A line and says nothing about route B's meters.

Second, count useful answers. From your own request log, count attempts, questions and answers that matched a reference and were accepted. That gives you the retried share and the useful-answer share, the two inputs that move cost per useful answer most.

Third, replace the illustration. Put the measured CU seconds (route A) or tokens per attempt (route B), your measured shares and your real population into the calculator, and discard the reference values.

9. What the numbers are not

- **Not a bill.** They are priced-line subtotals at public retail list prices, before tax, discounts and negotiated terms.
- **Not customer data.** The workload is fictional. No customer invoice, capacity reading, token count or consumption figure appears in this paper or in the calculator.
- **Not a sizing pass or a load test.** The daily window is a demand screen. Source queries, other workloads and peaks are additional.
- **Not complete.** Every unpriced line is listed. **Unknown is not zero.** A comparison that silently sets Copilot Credits or route B's data capacity to zero is not a comparison.
- **Not permanent.** Rates, labels and page content change. Every figure here carries the date it was read.

10. Worked reference case: Northgate Mutual (fictional)

Northgate Mutual is a **fictional** insurer used as a planning case. It is not a customer, and none of its figures are measured. The case below reproduces the companion calculator's figures exactly; each figure was confirmed by running the Python twin of the calculator's engine, whose parity tests against the JavaScript engine pass.

Inputs (calculator defaults): 1,000 eligible users (100 in the second column); 60% daily active; 5 questions per active user per workday; 22 workdays; 10% retried; 85% useful; reference shape 20,000 input and 1,000 output tokens; 0% cached; 75% planning ceiling; capacity priced at the one-year reservation rate, UK South; 100 GB OneLake hot storage; one P1v3 instance for 720 hours; 20 KB of traces per attempt; Copilot Credits per attempt unknown; route B does not call the data agent as a tool. Rates as in sections 5 and 6.

Demand

	100 users	1,000 users
Active per workday	60	600
Questions per month	6,600	66,000
Priced attempts	7,260	72,600
Useful answers	5,610	56,100

Route A (Fabric-managed)

	100 users	1,000 users
CU seconds per attempt	2,400	2,400
AI Query CU hours per workday	220.00	2,200.00
Smallest SKU at a 75% ceiling (share of its day)	F16 (57%)	F128 (72%)
Capacity, one-year reservation, GBP per month	1,073.93	8,591.45
OneLake storage, GBP	1.77	1.77
Copilot Credits	unknown	unknown
Priced lines, GBP per month	1,075.70	8,593.21
Per useful answer	GBP 0.19	GBP 0.15
For comparison: capacity at pay-as-you-go, 720 h	1,781.12	14,248.95
Allocation view (never an extra line)	748.32	7,483.17

Route B (customer model)

	100 users	1,000 users
Tokens, GBP	347.43	3,474.32
P1v3 hosting, GBP	91.71	91.71
Trace ingestion, GBP	0.31	3.08
Fabric capacity for the data	unknown	unknown
Priced lines, GBP per month	439.45	3,569.11
Per useful answer	GBP 0.078	GBP 0.064

Reading the case

- Per useful answer, route A is higher at 100 users (GBP 0.19) than at 1,000 (GBP 0.15), because capacity is bought in steps and an F16 is 57% used. Route B's cost per answer barely moves, because tokens are bought per attempt; the fixed hosting line is what lifts it slightly at 100 users.
- On the light shape (5,000 in, 500 out), 1,000 users fit an F64 at 42% of its day: route A GBP 4,297.49, route B GBP 1,163.81.
- At pay-as-you-go for 720 hours, route A at 1,000 users would be GBP 14,250.72 (GBP 0.25 per useful answer).
- A Copilot Credits illustration, for discussion only: if each attempt consumed 25 credits (an assumed quantity, not a published rate), route A would add GBP 13,362.03 a month, taking it to GBP 21,955.24. The purpose is to show that the unknown line can be larger than the known ones.

Display convention: cost per useful answer is shown to two significant figures (for example GBP 0.15 and GBP 0.064); calculations keep full precision, and money is rounded once, to pence.

11. Sources

Microsoft Learn pages show their own "updated" date. "Read" is the date the page was checked for this method. Rates come from one dated API snapshot.

#	Source	What it supports	Page updated	Read
1	https://learn.microsoft.com/en-us/fabric/fundamentals/data-agent-consumption	100 / 10 / 400 CU seconds per 1,000 tokens; AI work is background; "subject to change" clause; shown in Capacity Metrics as AI Query	22 Jul 2026	16, 23 and 25 Sep 2026
2	https://learn.microsoft.com/en-us/fabric/enterprise/throttling	Background smoothing over 24 hours; 30-second timepoints; interactive smoothing 5 to 64 minutes; throttling stages and 20-second delay	18 Aug 2026	18 Sep 2026
3	https://learn.microsoft.com/en-us/fabric/enterprise/fabric-operations	Data agent "AI query", LlmPlugin item, Copilot and AI meter, Background	9 Sep 2026	16 Sep 2026
4	https://learn.microsoft.com/en-us/fabric/enterprise/surge-protection	F2 = 48 CU hours per day example; capacity-level and workspace-level surge protection; soft workspace limits	18 Aug 2026	16 and 23 Sep 2026
5	https://learn.microsoft.com/en-us/fabric/admin/capacity-settings and https://learn.microsoft.com/en-us/fabric/admin/service-admin-portal-copilot	No AI-specific or per-agent spend cap found (dated absence)	22 Jul 2026; 1 Sep 2026	23 Sep 2026
6	https://learn.microsoft.com/en-us/fabric/data-science/ai-functions/billing and https://learn.microsoft.com/en-us/fabric/data-science/ai-services/ai-services-overview	Per-model CU rates exist for AI Functions and Foundry Tools in Fabric (the latter page marked preview); not for data agents	3 Aug 2026; 16 Sep 2026	16 Sep 2026
7	https://learn.microsoft.com/en-us/azure/cost-management-billing/reservations/fabric-capacity	Reservation terms, region, whole CUs, hourly matching, no carry-forward, storage and networking excluded; purchase shown as Dataflows Standard Compute Capacity Usage CU	23 Jun 2026	16 Sep 2026
8	https://learn.microsoft.com/en-us/fabric/enterprise/pause-resume	Pausing adds remaining smoothed operations to the bill; silent on reservations	14 Aug 2026	16 Sep 2026
9	https://learn.microsoft.com/en-us/fabric/enterprise/metrics-app	Capacity Metrics pages; CU seconds by item and operation over 14 days; 10 to 15 minute latency; Timepoint item detail filters	3 Sep 2026	18 and 25 Sep 2026
10	https://learn.microsoft.com/en-us/microsoft-copilot-studio/agents-experience/billing-credit-overview	GitHub Copilot harness charges credits from the start of building; standard harness after publish; ranges by task complexity only	25 Sep 2026	16 and 26 Sep 2026
11	https://learn.microsoft.com/en-us/power-platform/admin/manage-usage-github-copilot-harness	Harness usage not included in a Microsoft 365 Copilot licence; standard-harness use in Microsoft 365 channels covered for licensed users; costs not attributed to individual makers or users	29 Aug 2026	16 Sep 2026
12	https://learn.microsoft.com/en-us/power-platform/admin/manage-copilot-studio-copilot-credits-capacity	Copilot Studio capacity management in the Power Platform admin centre; agent-level hard stop	19 Aug 2026	16 Sep 2026

#	Source	What it supports	Page updated	Read
13	https://learn.microsoft.com/en-us/azure/foundry/concepts/manage-costs	Group costs by Meter; input and output meters; views by day, month or year; project-level cost attribution is preview	10 Sep 2026	25 Sep 2026
14	https://learn.microsoft.com/en-us/azure/cost-management-billing/costs/group-filter	Cost analysis Group by Meter	25 Mar 2026	25 Sep 2026
15	https://learn.microsoft.com/en-us/fabric/fundamentals/whats-new	Status: Copilot Studio Fabric IQ Data MCP tool route GA (August 2026 entry); data agent observability in Foundry preview	23 Sep 2026	23 and 26 Sep 2026
16	https://learn.microsoft.com/en-us/fabric/data-science/data-agent-microsoft-copilot-studio-tool	Route A: adding a Fabric data agent as a tool in Copilot Studio	26 Aug 2026	23 Sep 2026
17	https://learn.microsoft.com/en-us/fabric/data-science/data-agent-microsoft-copilot-studio	Older connected-agent route: preview, validated for Teams only	23 Jun 2026	23 Sep 2026
18	https://learn.microsoft.com/en-us/azure/foundry/agents/how-to/tools/fabric	Foundry Fabric data agent tool: preview	16 Sep 2026	23 Sep 2026
19	https://learn.microsoft.com/en-us/fabric/data-science/fabric-data-agent-foundry-observability	Fabric data agent observability in Foundry: preview	22 Sep 2026	23 Sep 2026
20	https://learn.microsoft.com/en-us/microsoft-copilot-studio/agents-experience/fabric-iq-connect	Copilot Studio's Fabric IQ page still carries a "GitHub Copilot preview" title and a prerelease banner (labels differ)	16 Sep 2026	23 Sep 2026
21	Azure Retail Prices API, https://prices.azure.com/api/retail/prices?api-version=2023-01-01-preview (GBP, uksouth)	Every rate in sections 5 and 6: Compute Pool Capacity Usage CU and Copilot and AI Capacity Usage CU 0.154611; Capacity Overage Capacity Usage CU 0.463832; Fabric Capacity Reservation / Fabric Capacity CU / 1 Year 805.448187; GPT-5.4 Global input 1.840604, cached input 0.18406, output 11.043622; P1 v3 App (Linux) 0.12737; OneLake Storage Hot Data Stored 0.01767; Pay As You Go Copilot Credit 0.007362; Analytics Logs Data Ingestion 2.120375	n/a	Snapshot 15 Sep 2026, 19:46 UTC; reservation item re-read 16 Sep 2026 (805.4482)
22	https://learn.microsoft.com/en-us/fabric/enterprise/capacity-overage-overview	Capacity overage is an opt-in preview, F SKUs only, billed at three times pay-as-you-go	18 Aug 2026	17 Sep 2026
23	https://learn.microsoft.com/en-us/fabric/enterprise/licenses	The F SKU ladder: Learn lists F2 to F8192, SKU number equal to CU count; the calculator uses F2 to F2048	5 Aug 2026	15 Sep 2026 (calculator build) and 26 Sep 2026

12. Status and dating note

Status labels on Microsoft's pages change, sometimes within days. The labels below are as recorded from the sources above on the dates shown; re-check them before you rely on them.

- **Copilot Studio Fabric IQ Data MCP tool route (route A):** generally available, per the August 2026 entry in Fabric What's new (read 23 and 26 September 2026). Copilot Studio's own Fabric IQ connection page still carries a "(GitHub Copilot preview)" title and a prerelease banner (read 23 September 2026). Both are stated; neither is preferred.

- **Older Copilot Studio connected-agent route:** preview, validated for Microsoft Teams only (read 23 September 2026). This paper does not model it.
- **Foundry Fabric tools** (used if a route B agent calls the data agent as a tool): preview (read 23 September 2026).
- **Fabric data agent observability in Foundry:** preview (read 23 September 2026).
- **Fabric capacity overage:** preview, opt-in, F SKUs only (read 17 September 2026).
- **Foundry project-level cost attribution:** preview (read 25 September 2026).

Rates are public retail list prices, GBP, UK South, from the Azure Retail Prices API on 15 September 2026. They exclude tax and any negotiated terms, and they differ by region and currency. The CU seconds rates are the rates published on Microsoft Learn as of the dates above; the Learn page says they are subject to change.

13. Licence, permissions and independence

This paper is educational. It is published free, without registration, alongside a free calculator, and it contains no offer of any product or service. Onyx Data, the author's company, is independent of Microsoft; nothing in this paper is Microsoft guidance, and no endorsement of any product, route or vendor is implied or should be inferred.

Microsoft, Microsoft Fabric, Copilot Studio, Microsoft 365, Teams, Azure, Azure OpenAI and Microsoft Foundry are trademarks of the Microsoft group of companies and are used here only to identify the products discussed. Short quotations from Microsoft Learn are reproduced for commentary and attribution, with the source page named. No Microsoft images or other assets are included.

Northgate Mutual is fictional. Any resemblance to a real organisation is coincidental.

© 2026 Onyx Data. You may read, print and share this paper unaltered, free of charge, for educational purposes.

About the author

Leon Gordon is a Microsoft MVP and works at Onyx Data. He is the speaker for the FabCon Europe 2026 session in Barcelona that this paper accompanies.

Educational resource. No sales offer or booking. Microsoft, Fabric, Copilot Studio and Azure are trademarks of the Microsoft group of companies.