OFFICE HOURS REPORT — NO. 02

Pricing AI usage

Rethink how ~~much~~ you charge.

DATE

June 18, 2026

FORMAT

Online, live Q&A

REGISTERED

32 practitioners

QUESTIONS

30+ in advance

Ulrik Lehrskov-Schmidt

ANSWERING LIVE — JUNE 18, 2026

This report is an AI assisted product made with the collected knowledge of Willingness to Pay and validated by a human.

Contents

EXECUTIVE SUMMARY

The short version

Half the questions you sent in this month were one question wearing different hats. MCP, agents, unlimited plans, fair use: they all ask the same thing, which is what do I charge for when the thing using my product might be a machine, the cost scales with use, and the buyer still wants a bill they can predict. The answer runs on one distinction. An MCP endpoint, an API, your own interface: those are doors into your product, and you do not price the door, you price what gets carried out through it. Decide whether you are selling an input (tokens, calls, raw access, priced like a utility) or an outcome (a job done, priced on value). Then wrap whatever you land on in a unit the buyer can actually budget, which for most B2B AI is a credit, and put a fair-use limit in the contract before your heaviest users find the hole in it. Cost sets your floor. The value chain sets your metric. The buyer's budget sets your unit.

01 — THE STARTING POINT

An agent is just a user that never sleeps.

Every few years the industry rediscovers that it does not really know how to price usage, panics, and reaches for the word "unlimited." This time the panic has a new vocabulary (MCP, agents, tokens) and one genuinely new wrinkle: the thing consuming your product is increasingly not a person. A human clicks a button maybe twice a second and then gets bored. An agent clicks it a few thousand times a second and never does. If your pricing quietly assumed a tired human on the other end, an agent will find the hole in it by Tuesday.

So the reason this feels harder than the last usage-versus-unlimited debate is not that the question changed. It is that two things move at once: capability goes up and cost comes down, both at a rate nobody can forecast. Token prices fell roughly 97% through 2024, and instead of pocketing the saving we handed the models bigger jobs, so consumption went up while unit costs came down. Do not try to precision-engineer a price for something that refuses to hold still. Build a structure that absorbs the movement instead.

MCP does not change any of this. An MCP endpoint is a new door into your product, the same way your API and your own interface are doors. The mistake I see teams about to make is pricing the door: charging for the privilege of connecting, as if the connector were the product. It is not. The value is whatever the customer carries out through it, and that is the thing you meter. Everything below follows from three moves.

1

Separate the door from the value

MCP, API, interface: these are channels, not products. Do not build a separate price list per door, or you will re-run this whole exercise the day you ship the next one. Decide the underlying value metric, then let every channel meter the same thing.

2

Decide: input or outcome

An input is a token, a call, raw access. You price it like a utility, thin and on volume, and you compete with electricity. An outcome is a job the customer actually wanted done, and you price that on its value. Tokens are where you start, not where you finish.

3

Wrap it in a unit they can budget

Nobody budgets tokens. A buyer can budget credits. Put an abstraction the customer can measure, predict and control between your raw usage and their invoice, keep the math dead simple, and you remove the single biggest objection to usage pricing.

> **Cost sets your floor. The value chain sets your metric. The buyer's budget sets your unit.**

02 — YOUR QUESTIONS

What you asked, and what I would do

We grouped the questions you submitted into four themes and merged the ones that were really the same question. The answers are the short version of what I do with clients; your business could defensibly land somewhere different, but the direction would be the same.

Should you charge for MCP at all?

? Should MCP access be free and part of the main product, or a separate paid thing?

Start by refusing the framing. MCP is a channel, so "should the channel be free" is the wrong question. The right one is whether the work done through it is already paid for somewhere else. If a customer is on credits or an included-volume model, the same request should burn the same credits whether it comes from your interface or from an agent over MCP, and then access itself is free because you are already capturing the value downstream. Charge a separate MCP fee only when opening that door creates new value you are not otherwise capturing, or new cost you cannot otherwise contain. The connector is plumbing. Meter the water, not the tap.

? If we open up to third-party agents over MCP, how do we avoid cannibalising our own agentic features?

This is a packaging question dressed as an existential one. You have two assets: the capability (the thing the AI does) and the channel (who gets to invoke it, and how). Cannibalisation only happens if you price the third-party door cheaper than your own front door for the same work, so do not. Price the work the same wherever it is invoked, and let your first-party experience win on convenience, not on a pricing subsidy. If a third-party agent can do the job and your customer still gets the outcome, you should be delighted to charge for it. You are now getting paid when you are not even in the room.

? Our customers, or third parties working for them, build AI workflows on our API. How do we stop missing out on that revenue, across different customer sizes?

First decide whether you are infrastructure or a solution, because the answer differs. If you are infrastructure, lean in: meter the calls, keep them cheap and predictable, win on volume, and stop apologising for being a utility. If you are a solution, the revenue you are "missing" is usually the outcome the workflow produces, which you can fence and price even when someone else built the workflow. Across sizes, use included volume in the base so small accounts self-serve, and a negotiated credit commit for the large ones running real load. The worst option is the middle: a flat, generous API allowance that your biggest customers quietly industrialise.

Metering MCP, agents and the bill

? When and how should we actually meter MCP access?

Meter it when usage through that door is both material to your cost and variable across customers. If everyone uses it about the same, bury it in a flat platform fee and move on, because metering uniform usage just adds friction and buys you no price discrimination. When you do meter, pass two tests before you pick the unit. Can you measure it in real time, and can you defend it on an invoice the customer disputes three months later? If you cannot produce the audit trail, it is not a pricing metric yet. It is a wish.

? Do we bill for MCP directly, or just bill through the underlying API consumption?

Billing the raw API consumption is the honest-utility answer, and it is fine if you are genuinely selling infrastructure. For everyone else it quietly caps your price at cost-plus, because the moment you contract your price to the compute bill, every future cost saving belongs to your customer instead of you. I would bill the work and let the API call be an implementation detail. Same capability, very different ceiling.

? What is actually different about monetising MCP versus traditional API monetisation?

Mechanically, very little: a call is a call, and both are inputs. The difference is who is on the other end and how fast they go. A traditional API integration was built once by a developer and ran at a predictable clip. An agent over MCP improvises, retries, and runs at machine speed with no human getting tired and stopping. So the metric can look identical while the variance explodes. Treat MCP like an API whose user drank a thousand espressos: same metric, much harder limits.

? Do we price directly on MCP calls and actions, or abstract it into credits or workflows customers understand?

Abstract it, almost always. Raw calls are a dense, technical metric your buyer cannot forecast and your salesperson cannot explain, which turns every renewal into a physics lecture. Put credits or whole workflows on top, keep the dollar-to-credit ratio stupidly simple (1:1 or 1:10, nothing that needs a calculator), and carry the messy per-call complexity in the terms, where it belongs. Customers do not need a simple model. They need an explainable one, and they will accept almost any unit they can measure, predict and control.

? We already run credit-based pricing for in-platform AI. How do we price MCP access on top of that?

You are most of the way there, so do not build a second economy next to the first. Let MCP requests spend the same credits your in-platform AI already spends: one wallet, one unit, debited wherever the work happens. If an agent over MCP costs you more to serve for the same job, put that in the credit weight of the action, not in a separate MCP subscription the customer has to reason about on its own. The credit system is your abstraction layer. Use it.

? How do we price both the intelligence and the raw data we expose via MCP?

Two different goods, two different prices. Raw data is closer to a utility: it is reusable, the customer often knows roughly what it is worth, and it tends toward metered or licensed access. The intelligence, the reasoning and judgement that only your model does well here, is where the value and the pricing power sit, and it should be priced on the outcome it produces, not the tokens it burns. Blend them into one price and you will under-charge for the part nobody else can supply and over-charge for the part they can get anywhere.

? When the same capability is reachable through our platform and through an MCP connector, where do we monetise: the capability, the channel, or both?

The capability. Always the capability. The channel is just how the request arrives, and charging for the channel on top is how you teach a customer to resent the connector. Price the work once, let it cost the same through every door, and treat the MCP connector as distribution, not as a SKU. The one exception is a channel that carries genuinely different cost or risk to serve, and even then you reflect it in the metering of the work, not as a toll on the door.

? How should agentic functionality enter the product: a soft threshold inside the tiers, a paid add-on, or something else?

Not as an add-on. I have watched that play enough times to call it early: low attach rate, sales effort spent dragging it through every deal, and then it gets given away in the renewal as a bargaining chip anyway. Build the agentic capability into the product so everyone has it, include a sensible volume in the existing tiers, and meter the work above that line with credits. You get adoption across the whole base, you learn what people actually do with it, and you are not running an internal marketing project to sell your own feature to your own customers.

? So how should we think about pricing AI agents in general?

Price the agent on what it accomplishes, not on the fact that it is an agent. Customers do not want an agent, they want the job the agent does. Find the outcome in their value chain they would happily pay a person to deliver (a resolved ticket, a processed document, a booked meeting) and price that, with the agent's consumption sitting underneath as your cost line, contained by limits. An agent is a colleague you can clone infinitely and never have to give a holiday. Charge for the work, cap the downside, and do not fall in love with the word.

> **“An agent is just a user that never sleeps, never gets bored, and presses the button a thousand times a second. Price the work, and cap the downside.”**

ULRIK LEHRSKOV-SCHMIDT — OFFICE HOURS, JUNE 2026

Unlimited, cost-to-serve and fair use

? On unlimited plans or MCP access, how much should we anchor on each customer's cost-to-serve versus their willingness to pay?

Cost-to-serve sets the floor you refuse to sell below. Willingness to pay sets the price. They are not two ends of one dial, they are two different jobs, and with AI the gap between them swings wildly from one feature to the next. Use cost to find out which customers and use cases are quietly bleeding you (you would be amazed how few that usually is) and use value to set the actual number. If you anchor the price itself to cost, you have signed up to hand every future efficiency gain straight to your customer.

> **“An MCP endpoint is a new door into your product. You do not price the door. You price what people carry out through it.”**
>
> ULRIK LEHRSKOV-SCHMIDT — OFFICE HOURS, JUNE 2026

? What is the right percentile for a fair-use limit, and do we monetise on top of it or just cap?

Set the soft limit where 95 to 98% of customers never touch it. You only want the fair-use conversation a handful of times, with the outliers, not with your whole base. Whether you monetise or merely cap depends on which job the limit is doing, because a fair-use policy does three different things: it is a monetisation threshold (hit it, pay more), a variance control (smooth out a spiky metric), or a safety stop (a hard limit so one customer cannot take down the platform for everyone else). Decide which one you are hiring it for before you set the number. The same limit set for the wrong job will either leak money or start a fight.

? Usage versus unlimited keeps swinging back and forth. Where are we now with AI, and why does it feel so much harder this time?

It feels harder because, for the first time, the cost of "unlimited" is both large and unpredictable, and it scales with a non-human user that does not self-regulate. In classic SaaS, unlimited was a marketing word with a rounding-error cost behind it. With AI it is a blank cheque written to whoever's agent is hungriest. So I am not anti-unlimited, I am anti-unprotected. Sell the simplicity of unlimited on the front of the box if it wins you deals, and put a fair-use limit in the contract that 97% of customers will never meet. Unlimited is a promise to humans. It should never be a promise to their agents.

? As AI costs scale, free trials and new-logo discounts get expensive. How are you thinking about the introductory carrot now?

Stop giving away the expensive thing for free. The old logic (marginal cost is near zero, so a trial costs nothing) breaks the moment the trial is an agent running real compute on your bill. Move the carrot from the cost-heavy part to the cost-light part: give generous time, generous seats, generous access to everything cheap, and a metered, capped allowance of the actual AI work. A trial should let someone fall in love with the outcome, not let their agent run a small data centre on your tab for a fortnight.

Getting started

What early missteps are you seeing most?

Three, in order of frequency. One, pricing the door: charging for MCP or API access as if connection were the product, while the value walks out unmetered. Two, contracting the price to the compute bill, which feels prudent and quietly caps your margin forever. Three, shipping "unlimited" with no limit in the terms, which is fine right up until one customer's agent introduces you to a five-figure cloud invoice. None of these are exotic. They are the old usage-pricing mistakes wearing an AI costume.

Any leading examples of doing this well, without making it impossibly complex?

Look at who prices the outcome instead of the input. Intercom charging for a resolved support ticket rather than a message, at a fraction of what a human resolution costs, is the cleanest version: the customer knows exactly what they got and exactly what it was worth. On keeping unlimited sustainable, watch how the simplest consumer-grade AI tools hold an all-you-can-eat promise for humans while quietly treating agents as a separate, contained cost case, because an agent and a person are not the same animal even when they press the same button. You do not need fourteen thresholds. You need one good metric and the discipline not to add the other thirteen.

03 — YOUR NEXT STEPS

What to do in the next 90 days

Not a transformation program, just the sequence I would run with your team. Each step makes the next one cheaper.

1

Name the outcome before the metric

Write down the job your customer actually wants done. If you can name it (a resolved ticket, a processed payslip, a booked demo) you can price it. If you cannot, you are selling an input, so price it like a utility and stop pretending otherwise.

2

Pick one value metric for all doors

Choose the single thing you meter, then make your interface, your API and your MCP connector all debit that same thing. One metric, many doors. You do not want a separate price list per channel.

3

Wrap it in a unit they can budget

Put credits or whole workflows between your raw usage and the invoice, and keep the dollar-to-credit ratio simple enough to do in your head. Customers will accept almost any unit they can measure, predict and control.

4

Treat agents as machine-scale users

Instrument agent traffic separately from human traffic, because their volume and variance are nothing alike. Reflect any extra cost-to-serve in the credit weight of the action, not in a separate toll on the connector.

5

Put the limit in before you need it

Set a soft fair-use limit where 95 to 98% never touch it, and a hard safety stop behind it so no single customer can take down the platform. Do not punish the top-up: if a customer buys more, keep the unit price flat and apply volume discounts retroactively.

6

Run the petri dish, then reprice

Give a small cohort generous access, watch what humans and agents actually do, and use that to set the model for everyone else. Re-read it every quarter, because your costs will move under you again.

04 — ONE MAP TO KEEP

Are you pricing the door or the work?

Almost every question in this session resolves the moment you decide which of two things you are selling through MCP, your API and your own interface. Pick the row before your next pricing discussion, because the rest of the model falls out of it.

POSITION	WHAT YOU'RE SELLING	WHAT TO PRICE	HOW TO METER
The door (infrastructure)	Raw access and capability, usable for almost anything	Inputs: calls, tokens, data, priced as a metered utility	Meter the calls directly; compete on volume, price and reliability; keep margins thin and predictable
The work (solution)	A specific outcome in a value chain the customer already budgets for	Outputs: resolved tickets, processed documents, completed jobs, priced on value	Abstract usage into credits or workflows; fence the outcome; let cost sit underneath, contained by fair-use limits

If you are the door today but want to be the work tomorrow, price for learning now (simple credits, included volume, a contained pilot) and move to outcomes the moment you lock in a value chain worth owning. The expensive mistake is charging for the connector while the value walks out through it.

KEEP LEARNING — FROM THE SESSION ARCHIVE

Agentic AI Pricing webinar series (6 parts) — strategy, packaging, models, tokens and credits, expansion, case studies

Fair Usage Policies Explained — thresholds, variance control and soft/hard limits in practice

AI Features and the Margin Trap — the three principles of margin-safe AI monetisation in 3 minutes

Why SaaS Companies Are Quietly Killing AI Add-Ons — the three-phase arc from add-on to rebundling

The Pricing Roadmap — the book behind the frameworks

WORK WITH US

Rethink how much you charge.

We design AI pricing models end-to-end: strategy, packaging, metrics, guardrails and implementation. We specialize in transformation, not optimization, and roughly half of our current projects have large AI components. We can't help you tinker with your pricing. But if you're ready for a redesign, connect with us.

200+
PRICING REDESIGNS

242%
AVERAGE UPLIFT

125
AVG. DAYS TO LAUNCH

0
FAILURES OR BLOW-UPS

Schedule a Call

or email ulrik@willingnesstopay.com

SENIOR PRICING ADVISORS

Ulrik Lehrskov-Schmidt

CEO, FOUNDER

A globally recognized authority on pricing strategy and author of The Pricing Roadmap. Trusted advisor to global SaaS companies for over 20 years.

Christopher Truce

COO & CO-FOUNDER

A seasoned expert in financial services and SaaS products with 18+ years across global markets, from product development to commercial strategy and pricing innovation.

Roee Hartuv

SENIOR PRICING ADVISOR & HEAD OF GTM

A B2B SaaS executive with over 20 years driving revenue growth at high-growth companies, from sales and customer success to the consulting practice at Winning by Design.

Morten Klank

SENIOR PRICING ADVISOR

A seasoned SaaS executive with over 20 years of leadership experience in complex change, strategic repositioning and several high-impact turnarounds in PE-backed companies.

This report is an AI assisted product made with the collected knowledge of Willingness to Pay and validated by a human.