



How should telcos price network APIs? A practical framework for driving adoption

Pricing remains one of the toughest questions in network API commercialisation. Different APIs, use cases and markets require different approaches, and setting prices too high can restrict adoption before value has been proven. This article sets out four recommendations to help telcos define viable prices and test them in the market.

Magnus Jeffery, Consultant

Network API pricing is becoming an adoption challenge

Network APIs have significant commercial potential for telcos; [STL Partners forecasts](#) that they could generate US\$31 billion in global telecoms revenue by 2030. However, commercial adoption has been slower than many in the industry expected. Coverage, integration effort and a shortage of proven use cases all play a part, but pricing is becoming one of the key barriers to adoption. Early network API buyers often highlight that high network API pricing was stalling commercial conversations before they had begun.

Often, telcos are perceived as trying to protect the theoretical value of their network data before customers have proved that value in production. There is also a tendency to anchor newer network APIs relative to SMS economics – either by applying a premium to SMS pricing, or by expecting APIs to compensate for declining SMS revenues.

However, customers do not assess an API based on the revenue stream it may replace for the operator. They assess whether it improves their existing workflow, and whether the economics make sense at the scale at which they would use it. Telcos need to set a pricing level that allows them to capture the value of their network capabilities without setting prices that restrict adoption and usage before the market has had a chance to scale.

A previous [STL Partners article](#) examined how the pricing model should vary by API and use case. The next question to address is: beyond the pricing model, what should the price be?

There is no single answer. Pricing will vary by API, market and use case, and learnings over the last six months have reinforced that even APIs within the same family can have very different value and volume profiles. However, there are four recommendations we believe should guide operators towards viable pricing.

1. Start with how the customer solves the problem today

Pricing considerations should begin with the customer workflow, rather than with the network asset itself. Operators need to understand what the customer is already using to solve the same problem – which may be another telco API, CPaaS API, a third-party data source, SMS OTP, device intelligence signals, OS location APIs, GPS, credit bureau data – or a combination of tools and manual processes. These existing solutions provide an important commercial reference point, because they establish what the buyer already pays and what level of performance they already receive.

The most relevant benchmark is therefore not necessarily another network API, but the product, process or combination of tools the API is competing with or complementing. A network API does not need to replicate an alternative exactly for that alternative to shape willingness to pay. Given the preponderance of data signals available on the market, comparisons with existing alternatives can therefore place downward pressure on network API pricing. Illustrative comparators show the range of economics buyers may already be accustomed to:

- [Twilio's public pricing](#) lists line-type intelligence at US\$0.008 per request and identity match at US\$0.10, subject to country variation.

- **Fingerprint** charges US\$99 per month for 20,000 device-identification calls, with additional usage priced at US\$4 per 1,000 calls. This includes a collection of device and risk signals rather than one raw data point.

It must be noted that these are end-customer prices, which include the margin of the provider selling to the buyer; a telco selling through an aggregator or platform needs to leave room for that layer.

An API that replaces an existing cost or removes complexity may have greater pricing headroom than one that simply adds another signal into an existing stack. SMS can therefore be a useful benchmark where it forms part of the customer journey being replaced (e.g. for Number Verification), but operators should avoid assuming that every network capability automatically warrants an arbitrary premium over SMS. Any premium needs to be justified by the incremental value the API adds to the customer's workflow.

2. Price for the API's contribution to the workflow, not the whole outcome

Network APIs often contribute to highly valuable outcomes without being responsible for the full value of those outcomes. An identity or fraud decision, for example, may combine several network APIs with other data signals (such as device intelligence, behavioural signals, transaction history and other identity sources). A network API may improve the decision, but it is rarely responsible for the entire outcome. Furthermore, the contribution can also vary significantly by API: Device Swap, for example, may provide an additional indicator of risk, whereas Scam Signal can provide more direct evidence of potential scam activity. Operators therefore need to understand the API's incremental role in the workflow: does it replace another paid input, remove a step, reduce failure or materially improve an important KPI? Value enabled by the overall solution is not the same as value the API itself can capture.

Pricing power is stronger where the API replaces another cost, removes complexity or demonstrably improves an important KPI – particularly where that impact can be isolated from the other inputs in the workflow. The more clearly operators can demonstrate this incremental impact, the greater the buyer's confidence and willingness to pay.

APIs that feed into revenue-generating use cases (e.g. KYC Fill In, Branded Calling) can support higher willingness to pay than APIs whose value is primarily cost avoidance. Where an API can increase customer conversion, completed transactions or paid sessions, the value is often easier to observe and attribute – and therefore easier for buyers to justify internally.

3. Price for the volumes you want to create

Price affects how often customers will use an API, and therefore the size of the opportunity. A high unit price may confine an API to a small number of high-risk transactions; at a lower price, the same capability may become viable to be embedded much more broadly across the customer workflow and vertical segments. A sufficiently cheap Number Recycling check would be used on every re-authentication check, but higher unit prices may restrict it to onboarding workflows for highly sensitive and high-paying verticals (such as financial services and healthcare). The question is

therefore not simply “what will a customer pay for one call?”, but “at what price does this API become economic at the frequency required by the target use case?”

One way to accommodate this is for unit costs to fall as usage grows. An API price that is acceptable during a small pilot can become uneconomic when applied to millions of transactions. Twilio's reassigned-number risk price, for example, falls from US\$0.02 at low volume to US\$0.0015 above six million queries – an over 10-fold difference. Pricing structure matters too. Monthly commitments, subscriptions and capped plans are often easier to approve than uncapped per-call charging, and outcome-based models are more attractive for customers where “success” can be clearly defined and measured.

4. Price for what the API is worth today, not the product it might become

Operators should be realistic about the maturity of the API when setting an initial price. Network APIs remain a relatively unfamiliar product category for many buyers, and operators and customers are still learning where these capabilities fit into their existing technology and decisioning stacks. This makes it difficult for operators to charge immediately for the full theoretical value of the capability.

A capability may ultimately justify a premium because of the uniqueness of the underlying network data, but buyers will only pay for differentiation that is understood, usable and proven in production today.

Gaps in operator coverage, inconsistent data quality, insufficient accuracy, high latency or uncertain service levels can materially reduce the value of an API. Customers are understandably reluctant to pay for unsupported numbers, inconclusive answers or failed verifications. In some cases, these are not simply reasons for a customer to pay less: they can make the API unsuitable for the workflow altogether. For example, if response times do not meet the latency thresholds customers are accustomed to from other data sources, the API may be unusable in real-time authentication or fraud decisions.

Operators should therefore expect initial pricing to reflect these limitations. Where coverage, latency or reliability are still below customer expectations, pricing may need to be discounted to encourage adoption and generate proof of value. As performance improves and the API becomes demonstrably more useful in production, operators can seek to capture more value through higher prices, stronger SLAs, richer data or broader propositions.

The right price has to be discovered in the market

The first, and most important, step that telcos should take is to test real prices in the market. Research and benchmarks can establish a starting hypothesis, but real customer usage provides the strongest evidence of what works. In an emerging market, getting a viable product into production and building customer adoption is more important than optimising the price to the last fraction of a cent before launch.

There is also a strategic cost to moving too slowly. If network APIs remain too expensive or difficult to adopt, customers and solution providers may continue building around alternative data sources and workflows, reducing the role of the telco in the long term. A modest paid deployment provides more evidence than another round of internal modelling.

Price discovery should operate as a continuous learning loop:

1. Set a price hypothesis for a defined use case and customer segment.
2. Run a paid pilot with agreed performance and volume assumptions.
3. Measure conversion, usage and the customer's business outcome.
4. Adjust the price, package or charging unit.
5. Repeat.

The pilot should test whether lower prices increase usage, whether volume commitments support adoption, whether customers will pay for better accuracy or coverage, and whether outcome-based charging reduces resistance. It must also distinguish a price problem from a product problem; low usage for technical reasons (e.g. latency issues) needs a different fix from low usage caused by price. Pricing should be treated as an iterative process, not a one-off calculation.

Operators should not conduct this price discovery alone. Network APIs often target customers, verticals and use cases that telcos have limited experience selling into directly. Distribution channels and enablement partners — including CPaaS providers, aggregators, solution providers and API platform providers — may have much closer visibility of end-customer economics, competing solutions and willingness to pay.

These partners should therefore play an active role in price discovery: sharing customer insight where available, helping structure pilots and testing whether pricing works in real workflows. This also helps operators understand the economics required across the full value chain, rather than setting a wholesale price in isolation.

Conclusion

Network API pricing should begin with the market operators want to create, not the theoretical value of the network asset. Telcos need to understand how customers solve the problem today, price for the incremental role the API plays in that workflow, and ensure the economics support the level of usage required for scale.

In an immature market, getting APIs into production and learning from real customer behaviour matters more than identifying a perfect price upfront. Operators that price too aggressively risk restricting adoption before the value of the API has been proven; those that create the right conditions for usage can build the evidence needed to capture more value as the market matures.

Magnus Jeffery is a Consultant at STL Partners, specialising in AI monetisation and Network APIs.

Get in touch with the author to learn more

magnus.jeffery@stlpartners.com

Or visit STL Partners' NaaS & Network APIs Hub

<https://stlpartners.com/naas-network-apis/>