



Pricing network APIs - matching the model to the use case

Network APIs create new ways for operators to monetise network capabilities, but per-call pricing will not suit every use case. Telcos need more flexible commercial models that reflect customer value. This article explores four approaches: subscription and tiered models, time- and session-based pricing, outcomes-based pricing, and proposition-led packaging.

Ella Barnard, Consultant

Why per-call pricing cannot be the default for network APIs

Network APIs create a new way for developers and enterprises to use telecoms networks, and a new route for operators to monetise capabilities that have historically been embedded within connectivity. Rather than treating the network simply as a blackboxed utility, they allow software applications to retrieve network-based information or request specific capabilities from the network. Examples include using Number Verification to confirm that a customer is in possession of a mobile device, Location Retrieval to establish the approximate location of a connected device, or Quality on Demand (QoD) to request differentiated network performance for a specific application or session.

Many telcos, hyperscalers, aggregators and developer platforms are now trying to answer a practical question around how network APIs should be priced. This is not a trivial question, because as the above examples demonstrate, network APIs do not behave like a single product category from a commercial perspective. The right pricing model therefore depends on what the API does, how the customer uses it, and where the value is created.

Historically, many APIs have been priced in the same way: per 'call', meaning the buyer pays each time an API is invoked. This is a familiar model for operators, reflecting the CPaaS and SMS approach where each message or transaction is charged individually. For early network APIs such as Number Verification or SIM Swap, it can also make sense, as the customer asks a discrete question and receives a discrete answer.

However, relying too heavily on a single per-call model creates several problems.

1. First, it scales cost directly with usage, which discourages adoption. Paying a charge per call may be acceptable in early pilots, but if every additional user, device or transaction creates a separate charge, customers may limit usage to the minimum rather than designing and innovating around the API at scale.
2. Second, it assumes that all API calls have similar value, when they do not. Customer willingness to pay varies by use case, vertical and market. QoD is a clear example: the same underlying capability could support a low-value gaming enhancement or a mission-critical remote maintenance workflow where avoiding downtime is worth significantly more.
3. Third, it risks commoditising APIs before the market has matured. If telcos expose raw capabilities on a horizontal, one-size-fits-all basis, value may be captured by aggregators or solution providers higher up the stack. STL's research highlights that APIs are often not bought as isolated product capabilities: For example, Location Retrieval is commonly embedded in broader propositions, while QoD should be packaged around use cases rather than sold as a generic API.

Per-call pricing will remain part of the toolkit, but it cannot be the default answer for every network API. As the market matures, operators will need a more flexible approach to commercialisation.

Four commercial models telcos should consider

Successful commercialisation will require telcos to make a few related but distinct choices: what the customer is charged for, how the price is set, and whether the API is sold standalone or bundled into a wider proposition. These choices are not mutually exclusive. In practice, one API does not have to mean one pricing model: operators may need to combine approaches, or offer different options depending on the customer segment, channel and use case. The aim should be to match the commercial model to how the customer uses the API and how value is created.

a. Subscription and tiered models: Making recurring API usage easier to scale

A first alternative to per-call pricing is the subscription model.

Subscription models can take several forms. At the simplest end, a customer pays a fixed monthly fee for access to an API, sometimes with a fair usage cap. This creates predictable revenue for the telco and predictable cost for the customer. A more structured version is tiered subscription pricing, where different packages include different usage volumes, service levels, support levels or geographic coverage. For example, a developer might buy a 'starter' tier for testing, a 'growth' tier for moderate production use, and an 'enterprise' tier for large-scale deployment.

Volume-tiered models sit somewhere between per-call and subscription pricing. The customer still pays based on usage, but the marginal cost falls as usage increases. This can help avoid penalising scale while still preserving a link between consumption and revenue.

These models work particularly well where customers want cost predictability and where API usage is expected to be regular, repeatable and embedded into ongoing operations. Location Retrieval is a good example. STL's research found that operators should consider subscription models for Location Retrieval monetisation, particularly device-level and subscription pricing models that align with enterprise buying behaviour. In STL survey results, 40% of respondents preferred per-device-per-month pricing for Location Retrieval, and 25% preferred the API to be included as part of a wider platform or software subscription.

This makes sense in practical use cases. A fleet management provider does not want to think about every individual location lookup as a separate commercial event; they want to manage a population of vehicles. A per-device-per-month model maps more naturally to how that business already sells and buys software.

KYC-related APIs may also move towards volume tiers or subscription models as adoption grows. Per-transaction pricing is intuitive at the start, particularly where a bank or digital service provider is using the API for a discrete verification event. But as usage becomes more embedded into fraud prevention, onboarding or compliance workflows, buyers may prefer predictable monthly pricing or committed-volume tiers. STL's

research has found that KYC may start per-call, but buyers often shift to volume tiers or subscriptions as usage grows.

The limitation is that subscriptions need to be designed carefully. For APIs where the operator's costs are relatively predictable, subscriptions can reduce friction and support adoption. But for APIs that draw on scarce or active network resources, an unlimited model may expose the operator to rising costs without a matching increase in revenue. This is where other models, such as time-based pricing, become more relevant.

b. Time- and session-based pricing: Charging for temporary and on-demand needs

Time-based pricing means charging for the period during which a network capability is active. This could be per minute, per hour, per day or per session. It may also vary by 'flavour' of service, such as latency, reliability, uplink priority or guaranteed bitrate.

This model is especially relevant for QoD. Unlike a simple information lookup, QoD draws on active network resources for the period in which enhanced performance is needed. In most cases, that makes time-based pricing a natural fit: the customer pays for a defined task, session or workflow, rather than for an always-on capability. It is also relatively familiar to enterprise buyers, as it mirrors the way many already consume cloud compute services.

A drone inspection is a useful example. The customer does not need enhanced performance all day, but for the duration of a flight, or for a specific stage of an inspection when video upload, control signals or safety requirements are most critical. An hourly or session-based QoD charge is more intuitive than charging per API call.

The same logic applies to live broadcasting. A media company covering a sports event may need assured uplink for a defined production window. A temporary, time-based QoD package could be easier to sell than a complex usage-based model.

Time-based pricing can also be combined with performance profiles. STL's QoD work highlights the need for profiles that define both the technical characteristics, such as uplink priority, latency consistency, reliability and guaranteed bitrate, and the commercial dimension, reflecting customer value and willingness to pay. It also notes that QoD cost can vary by capacity, congestion, geography and network readiness.

In practice, this suggests a relatively simple principle: charge for time, but vary the price according to the performance profile, location and use case. A basic gaming enhancement should not be priced like a mission-critical industrial maintenance session.

That said, time-based pricing is still largely a consumption-based model. It tells the customer what they are paying for in terms of duration and quality, but not always in terms of the business result. For some use cases, the next step is to link pricing more directly to the outcome achieved.

c. Outcomes-based pricing: Charging for a successful or measurable result

Outcomes-based pricing links payment to the result achieved. In practice, this could mean charging only for a successful verification, a confirmed match, an avoided failure, a completed transaction, or a guaranteed service outcome.

The appeal is clear: customers are understandably reluctant to pay for API calls that do not return anything useful. In identity and fraud use cases, for example, a buyer may object to paying the same amount for a 'no match', an inconclusive result and a confirmed answer. This was borne out at the [recent Aduna summit](#), with buyers such as Meta, Google Firebase and LexisNexis arguing in favour of API pricing models that enable them to link payment directly to business value, including per-verified or per-successful-outcome models.

The challenge in delivering outcome-based pricing is measurement; it requires proof of delivery, clear attribution and a shared definition of success. To make this work, operators need to be able to define what level of service is being promised, check whether the customer or application is eligible to receive it, and prove that it was delivered. Without that, it becomes harder to charge for improved network quality, or to resolve disputes if the customer does not believe they received what they paid for.

For that reason, outcomes-based pricing is unlikely to replace all other models. It is more likely to work in selected use cases where the result is valuable enough to justify the extra commercial complexity. In other cases, the better route may be to price and package the API as part of a broader proposition, where the customer pays for the overall solution or workflow improvement rather than the API outcome alone.

d. Proposition-led pricing: Packaging APIs into broader offers

Proposition-led pricing means the customer does not necessarily buy a single API as a standalone product. Instead, the API is packaged in a way that reflects the customer problem or use case.

This can take two main forms. The first is bundling multiple APIs that are more valuable together than separately. For example, an identity or fraud proposition might combine Number Verification, SIM Swap and KYC Match into a broader trust or authentication package.

The second is embedding an API into a wider solution, software platform, connectivity package or managed service. In this case, the API is an enabler of the proposition, not necessarily the product the customer thinks they are buying.

This is likely to be important because many enterprise buyers do not want to procure raw network capabilities. They want outcomes - perhaps fraud prevention, fleet visibility, safer drone operations, better live video contribution, or more reliable point-of-sale connectivity at a venue.

Location Retrieval illustrates this model well. It can be embedded into a logistics platform as a fallback to GPS, or into a workforce safety application to verify that field workers are in the expected area. In both cases, the end customer may pay for the application or workflow improvement, not the API itself.

The same applies to QoD. A venue services provider might buy a managed package that includes baseline connectivity, temporary QoD boosts for payment terminals, and support during major events. A remote maintenance provider might include QoD as part of a premium service tier, rather than asking customers to buy it separately.

For telcos, the commercial implication is significant. Proposition-led pricing can move network APIs higher up the value chain, but only if telcos have the right channels, partners and packaging capabilities. Otherwise, aggregators and application providers will package the APIs into higher-value offers and capture more of the upside.

No single pricing model will fit every network API

There is no single correct pricing model for network APIs. Per-call pricing remains useful for simple, discrete lookups, particularly in early-stage adoption, but it is not sufficient to grow the commercial opportunity that network APIs present to telcos.

The practical lesson is that telcos should not start with the API and then apply a default price. They should start with the use case, the customer's buying behaviour and the value created. Different APIs, and even different use cases for the same API, will need different charging units, packages and price levels depending on the scale of usage, how predictable demand is, whether the API consumes active network resources, and how directly it contributes to the customer's business outcome. Building this flexibility into pricing will be important for operators to maximise commercial opportunities as the market matures.

Ella Barnard is a Consultant at STL Partners.

Get in touch with the author to learn more

ella.barnard@stlpartners.com

Or visit STL Partners' NaaS & Network APIs Hub

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