Response to BEREC Call for Inputs on Interfaces for Mobile Networks for Developers and Third-Party Services

Dean Bubley, Director, Disruptive Analysis

Executive summary

I am broadly supportive of the development of standardised network APIs, including via CAMARA and GSMA Open Gateway initiatives. The previous situation, where individual operators exposed proprietary interfaces with different semantics, commercial models and developer processes, was clearly unsatisfactory, except for very narrow niches such as SMS messaging.

However, there is a risk that discussion of network APIs becomes too supply-driven and too narrowly focused on public mobile networks / MNOs.

The existence of an API specification, or even an operator implementation, should not be confused with developer adoption, commercial demand or sustainable revenue. BEREC notes that 228 API implementations are deployed or supported by European operators, while correctly acknowledging that this figure counts the same APIs multiple times across operators.

It should be acknowledged upfront that many network APIs are consumed inside operators themselves, for example between different business units (eg wholesale & retail, or mobile/fixed arms), by internal software development teams, or by product teams using them to create higher-level products and solutions which embed certain API functions within them. Arguably the internal MNO/telco opportunity is greater than that for external developers and third parties.

There is a second and increasingly important problem with the framing. A developer generally does not care whether an application or device is using 4G, 5G NSA, 5G SA, Wi-Fi, a neutral-host indoor system, private 5G, satellite direct-to-device connectivity or some combination of them. Increasingly devices will have multiple connections, often used simultaneously. The application cares about an outcome such as identity verification, adequate upstream bandwidth, low latency, privacy or trustworthy location. Developers will also usually seek to cost-optimise across multiple domains, including direct API costs, deployment and testing time and resources, skills requirements, lock-in to particular technologies or operators etc.

The more an API relates to hard measures or control of **network performance**, rather than subscriber information, the harder it becomes to regard "the mobile network" as a single controllable entity.

My main conclusions are therefore:

1. **Network APIs are useful, but they are a contract surface rather than a new economic moat.** Standardisation removes friction. It does not by itself create demand.

2. **There is a major difference between APIs exposing unique telecom data and identity capabilities, and APIs attempting to monetise network performance.** Number Verification, SIM Swap, device status, fraud signals and some location capabilities already address identifiable problems. The commercial case for Quality on Demand, network slice booking and optimal-edge discovery is less established, and has huge dependencies on network coverage and other factors.
3. **The relevant future architecture is cross-network rather than mobile-only.** Applications increasingly span cellular, Wi-Fi, fixed broadband, private networks, neutral hosts, indoor systems and satellite connectivity. CAMARA itself is already moving beyond a purely cellular conception, including APIs covering connectivity insights, dedicated networks and communication services.
4. **BEREC should measure consumption rather than publication.** Relevant metrics include paid production API calls, active applications, recurring enterprise customers, revenue, cross-operator implementations and developer retention, not the number of specifications, participating operators or demonstrations.
5. **Interoperability is much more than common API syntax.** Two operators can implement the same CAMARA API while providing different underlying functionality, geographic coverage, indoor performance, QoS characteristics, consent flows, reliability, commercial terms and error behaviour.
6. **Device and subscriber identity are becoming more complicated.** Multi-SIM, eSIM, travel eSIM, enterprise eSIM and data-only subscriptions create cases where the device, phone number, currently active data subscription and subscriber identity are not synonymous.
7. **Performance APIs depend on information that operators may not possess with sufficient granularity.** QoD, slice booking and edge placement can require detailed knowledge of actual radio coverage, current and predicted capacity, indoor conditions, local Wi-Fi and neutral-host infrastructure and the path between the user and application. A slice booked for a journey would need mapping between route and network coverage – perhaps even down to which lane of a road is being used.
8. **Consent is not a binary checkbox.** Location and other personal-data APIs raise questions about purpose, duration, delegation, revocation and whether consent applies per call, application, session or service. IoT adds cases where the device itself has no meaningful user interface.
9. **The market is likely to be mediated and productised.** Most developers will consume capabilities through CPaaS providers, cloud platforms, fraud platforms, vertical solution providers and, increasingly, AI agents rather than contracting independently with dozens of operators.
10. **AI agents create both an opportunity and a competitive threat.** Agents may become significant machine-speed consumers of network APIs, but they can also choose Wi-Fi, another SIM, another network, a different codec, local processing or an alternative fraud signal instead. They may also use direct connection to other AI systems via techniques such as MCP, rather than using conventional APIs.
11. **Network-information APIs may substitute for network-control APIs.** If an application learns that a cell is uncongested and already meets its requirements, there is no reason to buy QoD. This has important consequences for forecasts that simply add the theoretical addressable markets of individual APIs.
12. **Open APIs should not be confused with open markets.** A technically open specification can still lead to concentrated aggregation, restrictive commercial access or switching barriers.

13. **Voice, messaging and network APIs should be composable even if they remain technically distinct.** Developers increasingly buy outcomes such as a secure customer interaction rather than categories such as CPaaS, network API or WebRTC API.
14. **Regulation should initially concentrate on interoperability, transparency, consent, security, switching and competition rather than mandating a large catalogue of APIs or particular business models.** This remains an immature market and premature regulatory specification could fossilise the wrong architecture.

The main objective should therefore be to create a competitive and interoperable market in **communications and network capabilities**, rather than maximising the number of APIs.

An important framing issue: there is no single "network"

BEREC's Call for Inputs understandably focuses on interfaces to mobile networks. However, many of the applications invoked as justifications for advanced network APIs will not operate wholly within a single MNO-controlled environment, or a federated / aggregated set of MNOs.

The end-to-end path may include:

Connectivity domain	Implication for network APIs
4G	Identity and subscriber-information APIs can work, but fine-grained programmability depends on the operator's core and policy architecture.
5G NSA	The radio may be 5G while much of the control architecture remains anchored in 4G/EPC. API availability cannot simply be inferred from a "5G" indicator.
5G SA	Offers a richer basis for slicing and advanced QoS, but coverage remains uneven and a device may move between SA, NSA and 4G during a session.
Operator-controlled Wi-Fi	The fixed/mobile operator may have visibility of gateways, access points and home devices and potentially expose Wi-Fi-related information or control.
Third-party Wi-Fi	An MNO may have little or no visibility of radio congestion, LAN conditions or upstream connectivity. Authentication may not be automated, and various middle-box functions may break APIs.
Neutral-host cellular	The customer sees their normal MNO service, but part of the physical infrastructure may be owned and operated by a venue or neutral-host provider. In some cases, the NH may have its own spectrum and core, with a relationship based on roaming or MOCN sharing, for which network APIs are mostly poorly designed.
Indoor DAS/small-cell systems	Performance may depend on infrastructure sitting between the MNO core and end user over which the MNO has incomplete operational control & visibility.

Connectivity domain	Implication for network APIs
Private 4G/5G	Enterprise or private-network operators may control identity, radio, policy and local edge resources rather than the public MNO.
Satellite D2D / NTN	The radio path and capacity may be supplied by a satellite partner while subscriber identity and service remain associated with the terrestrial operator – although other D2D service may issue their own eSIMs or work with a rival or international MNO.
Multiple SIMs/eSIMs	The phone number relevant to an application may not correspond to the subscription currently carrying its packets. There is huge scope for false-positives / negatives eg for anti-fraud checks while using a Travel eSIM
VPN / ZTNA / SASE	The access operator may know the device and radio conditions while losing visibility of the application's actual endpoint or individual traffic flows.

This distinction becomes increasingly important as APIs move from relatively static information such as SIM-swap history toward promises about network performance. A SIM Swap API can answer a question about a subscription without knowing the quality of the user's current Wi-Fi connection. But a QoD API cannot meaningfully promise an end-to-end application outcome if a congested enterprise Wi-Fi network, poorly engineered indoor system, neutral host, VPN concentrator or remote cloud path is the actual bottleneck.

The correct conceptual object is therefore increasingly the **connectivity path**, rather than simply "the mobile network".

---

1. What roles do the different players have in the ecosystem?

The ecosystem should be considered as several overlapping layers rather than a simple relationship between an operator and a developer.

Mobile operators

MNOs are important capability providers. They expose information or control functions derived from networks, subscriber relationships, policy systems, security platforms and OSS/BSS. They may also consume APIs internally, for instance in product-development teams or their own retail operations.

Their most important responsibilities are not API development itself, but ensuring that an exposed capability is:

- accurate
- properly authorised
- operationally reliable
- geographically and technically applicable
- explicit about its limitations.

The last point is important. An operator should not imply that it controls an end-to-end outcome when it controls only part of the path.

MNOs are also increasingly involved in various industry-wide initiatives, aggregation roles and sector-specific collaborations eg on banking and payments anti-fraud. Some also have equity roles in adjacent companies such as Aduna.

MVNOs, MVNEs and alternative SIM providers

MVNOs and MVNEs require particular attention. They should not automatically be excluded from network-API economics because they do not own the underlying RAN. There are specific competition issues here that regulatory authorities should consider.

Depending on the specific capability and contractual model, they may own:

- a 3GPP core network
- the customer relationship
- the phone number
- billing
- consent
- enterprise integration
- security policy
- eSIM provisioning
- application distribution

Travel-eSIM providers add another layer. A user may retain their normal home number for banking, messaging and account recovery while using an unrelated travel eSIM for current mobile data.

This makes the question "which subscriber is using this device?" surprisingly ambiguous.

At present, MVNOs have either been excluded from many Network API initiatives, or have had minimal focus. Most are not GSMA members, so may have only limited engagement with efforts such as Open Gateway.

Wi-Fi, fixed and indoor connectivity providers

The network API ecosystem should explicitly recognise Wi-Fi and indoor wireless providers.

This includes:

- fixed broadband operators controlling home gateways and Wi-Fi
- enterprise Wi-Fi
- venue Wi-Fi
- managed Wi-Fi providers
- neutral-host cellular networks
- distributed antenna systems
- shared small-cell systems

CAMARA has already explored APIs extending QoS into home Wi-Fi, and its Session Insights work explicitly contemplates Wi-Fi congestion as a cause of degraded application performance. Some of the recent anti-fraud APIs such as silent authentication can now work when the user is connected to Wi-Fi, in a similar fashion to Wi-Fi calling.

The fact that an API can theoretically optimise the MNO part of a connection does not mean it can optimise the user's complete experience.

Mobile-centric groups should work closely with their peers in other sectors such as WBA, Wi-Fi Alliance, Broadband Forum, Mplify and CableLabs to maximise alignment wherever possible.

Private network operators

Private networks complicate the model further. An industrial enterprise may own or control:

- the local 5G core
- spectrum or local licence
- radio infrastructure
- edge compute
- identity and security for employees, visitors or IoT devices
- SIM/eSIM estate
- policy management

CAMARA's Dedicated Networks APIs already recognise the idea of reserving network resources, selecting capabilities and controlling device access within defined areas and times. However, many enterprises and their application developers may have other considerations, such as managing their own multiple tenants. They may also want to use other techniques for managing virtual resources, rather than 3GPP-style "slices."

Additional complexities arise in hybrid networks that blend MNO and Private 5G capabilities, or where users or devices roam between public and private domains, perhaps with separate identities or eSIMs.

A future developer may therefore need a capability that works across public MNO infrastructure and an enterprise-controlled private network.

See this documents (written for a client) for a more detailed discussion:
https://shabodi.com/whitepapers/network-api-a-new-use-case-category-for-private-wireless/
https://shabodi.com/whitepapers/enterprise-network-apis-more-than-just-5g-duplicate-32616/

Satellite operators

Satellite D2D and NTN should also form part of the BEREC Network API analysis. A future mobile user may remain commercially associated with an MNO while temporarily receiving access through an external satellite operator. The customer may regard this as the same service, while radio conditions, capacity and geographic availability are determined by an entirely different infrastructure provider.

An API asking "is this device reachable?" may still make sense. There should also be consistent behaviour with anti-fraud and location APIs, although that may be complicated based on the ground station location and network topology.

Satellite operators may also use alternative eSIMs, either from a rival MNO (eg see what T-Mobile is doing in the US, or Rogers in Canada), or associated with another party such as a travel eSIM, or perhaps even a "pure" satellite eSIM in future, depending on spectrum and terrestrial RAN assets.

However, an API claiming to provide a particular QoS profile becomes much harder if the device transitions onto a satellite link that cannot support it. At the moment, there does not appear to be a set of CAMARA extensions for D2D services.

Aggregators and CPaaS providers

Aggregators, CPaaS providers, channel partners and cloud platforms are highly important for some network API consumption scenarios, but less relevant for others.

Consumer-centric mobile apps need to work across most / all MNO networks in a given country. This has been central to some successful implementations such as anti-fraud APIs for banking transactions.

Conversely, some network API scenarios will be less driven by multi-network aggregation. For instance, if a retail chain deploys a 5G fixed-wireless backup connection at each of its stores, using a high-availability, QoD-powered hybrid SDWAN solution from a specific MNO and its fixed-network arm, it is less important that the APIs are available on other MNOs as well.

However, aggregation also creates concentration risk. The industry should be wary of one or two dominant aggregation gateways or partnerships. There is also an important regulatory and competitive consideration around MVNOs and smaller MNOs, which may not get representation via aggregation platforms.

Vertical API-enabled solution providers

There is also a category that BEREC should make more explicit: **vertical solution providers that use network APIs as ingredients rather than products**.

Shush is one example in fraud prevention and network authentication. Its Sherlock platform describes itself as a CAMARA-compliant API gateway and network-authentication platform incorporating Number Verification, SIM Swap, Device Status and other network attributes into solutions for operators, CPaaS providers and enterprises. It therefore packages carrier data, business logic, integration and commercialisation around the underlying APIs.

This is potentially a more important commercial model than developers buying individual APIs directly.

The value chain becomes:

Network capability → standard API → vertical intelligence/rules → workflow/product → enterprise outcome

Examples could include:
- fraud platforms
- drone-management systems
- automotive platforms
- live-production connectivity systems for broadcasters
- industrial automation vendors
- healthcare platforms.

BEREC should therefore distinguish **network API revenue** from the much larger potential revenue of products containing network APIs, to avoid either double-counting or overstated numbers.

AI agents

Finally, AI agents should explicitly be added to BEREC's model of the ecosystem. They are not merely another type of developer or end-user.

An agent may dynamically:

1. understand an application intent
2. discover available network or communications tools
3. inspect current network conditions
4. compare alternative ways of fulfilling the intent
5. invoke one or more APIs
6. evaluate the result
7. change strategy if the result is unsatisfactory.

This makes an agent both a **consumer and an orchestrator** of APIs.

This area was explored in the January 2026 Unthinkable Lab event and subsequent webinar run jointly by Disruptive Analysis, Connective Insight and TelecomTV:
https://www.telecomtv.com/content/unthinkable-lab-agentic-ai-plus-telecoms-apis/ (see webinar recording halfway down the page)
See also the output report: https://www.telecomtv.com/content/unthinkable-lab-agentic-ai-plus-telecoms-apis-report/unthinkable-lab-report-accelerating-telco-apis-with-agentic-ai-54756/

---

2. Market demand and development

BEREC should be cautious about network API market forecasts.

Many forecasts effectively extrapolate a theoretical addressable market from application categories that *might* benefit from network information or programmability. That is very different from demonstrated willingness to pay.

There are at least four distinct API markets.

Identity, authentication, security and fraud - Number Verification, SIM Swap, Device Swap, KYC-related capabilities, Account-security signals. These solve existing enterprise problems and can enhance or substitute for existing verification or fraud-prevention mechanisms. Banks, retailers, airlines and others with large-value transactions may pay for these.

Network and device information. Location, roaming state, reachability, connected network type, current connectivity quality, predicted connectivity. These may be valuable where operator information is uniquely authoritative or difficult to reproduce elsewhere. However, some of these, such as predicted connectivity, may require significant compute resource and cost. "Will this network work at place X at time Y?" is a hard problem to solve reliably. Some APIs such as location and roaming may have consent and privacy concerns.

Network control and programmability. Quality on Demand, QoS booking, network slice booking, dedicated networks, edge compute, traffic influence. This category has some of the most ambitious commercial expectations and some of the most difficult implementation dependencies. There is read-across to having full visibility of granular network coverage, including indoors. There is also an issue around inventory management and capacity – for example, how many uplink-guaranteed video streaming sessions can be supported in a football stadium during a match? What happens when the QoD network capacity is "full", and what happens to normal non-priority users in that situation? How are complaints dealt with, and is there a regulatory reporting mechanism for failures, especially if there are potential harms such as accidents involving machinery or vehicles?

Communications services. CAMARA's current portfolio also includes communication-service capabilities such as WebRTC call handling, registration and event subscriptions. This begins to overlap with the territory historically occupied by CPaaS, SIP, programmable voice and messaging. The market should therefore not be assumed to divide neatly into "network APIs" and "communications APIs".

Network information can cannibalise network control

This deserves much more attention in market forecasts. CAMARA's Connectivity Insights API explicitly contemplates two different actions when application requirements are not or may not be met:

- request a network boost through QoD
- change the application's own configuration, such as video resolution.

This creates an important substitution effect. Suppose an application wants 20 Mbit/s uplink for a live video stream. If a network-information API reports that the cell is lightly loaded and current performance comfortably exceeds 20 Mbit/s, there is little reason to purchase QoD. If conditions deteriorate, the application might:

- purchase QoD
- reduce resolution
- change codec
- buffer temporarily
- move onto Wi-Fi
- use a second SIM
- postpone transmission
- process more data locally

Only one of those outcomes produces QoD revenue. Predictive Connectivity Data potentially expands the alternatives further. CAMARA describes the API as estimating future 4G/5G connectivity for a geographic area, time and even height using network coverage and cell-load data. That can enable an application to route around poor connectivity rather than pay to modify it. For instance, a drone might change route, a vehicle might pre-cache content or delay an OTA update, or a backup job might wait ten minutes.

Information and control APIs are therefore partly complements and partly substitutes. This has significant implications for market sizing. Forecasts should not simply calculate:
Connectivity Insights TAM + QoD TAM + Slice Booking TAM + Edge Discovery TAM because the APIs may compete amongst themselves to solve the same underlying problem. They could even reduce the overall operator revenues in some circumstances.

This also suggests that some advanced network-control API markets may develop later and be smaller than current top-down forecasts imply, while information APIs may appear earlier.

---

3. Business and pricing models

There is unlikely to be just one successful pricing model for network APIs. Transaction pricing makes sense for capabilities such as identity verification or fraud checks because an API request corresponds reasonably closely to a business event. Subscription and tiered-volume pricing may suit enterprise monitoring and information APIs.

However, performance-related APIs are different. Charging per QoD API invocation risks pricing the mechanism rather than the value. Outcome-oriented pricing may therefore eventually be more appropriate, perhaps delivered via the MNO creating a complete product / solution, using its own APIs internally. For instance, an MNO-marketed drone-as-a-service proposition, or the aforementioned 5G FWA SDWAN backup service might use multiple internal security and network APIs.

However, outcome pricing transfers more risk to the provider. If an application pays for assured performance, someone needs credible measurement, SLA definitions, attribution of failure and possibly refunds. Part of the word "reliability" is "liability".

Diagnosis and attribution of failures will also be complex here, especially if there are multiple possible "culprits" such as the MNO RAN, the core network, an indoor neutral host, the application's logic or video codec, a VPN gateway, a cloud resource etc.

BEREC should not attempt to determine preferred pricing models. It should instead encourage commercial transparency and ensure that wholesale structures do not create unjustified barriers to competition or switching.

Voice, messaging and network APIs

There is also an important market-definition question around whether network APIs will be sold separately from programmable voice and messaging. Technically, many should remain separate, as they may have different security requirements, operational characteristics, charging models and regulatory obligations.

Commercially, however, keeping them in separate silos makes progressively less sense. A bank does not want "a SIM Swap API". It wants a safer customer interaction. That product might combine Number Verification, SIM Swap, Verified Caller, Voice Biometrics, Messaging via MNO or App channels, Device Intelligence, 3rd-party Fraud Scoring.

My expectation is therefore: **common commercial discovery and productisation, but continued separation of lower-level technical and security domains.**

BEREC should avoid defining a market so narrowly that network APIs appear economically independent from CPaaS and broader communications services when, in reality, customers may increasingly consume them together.

---

4. How can open APIs support a dynamic European market?

Standardised APIs can reduce integration costs and increase the addressable market for an application. That is clearly positive, but "open API" requires more careful definition.

CAMARA specifications may be openly developed while access to production implementations remains subject to private contracts, commercial restrictions, operator-specific availability, consent requirements and network-specific implementation limitations.

Technical openness is therefore not equivalent to open market access. A healthy European market should support portability between aggregators, multiple competing distribution channels, consistent identity and authorisation mechanisms, reliable test environments, reasonable switching between providers, equivalent treatment of competing applications, participation by MVNOs and participation by private and neutral-host infrastructure providers where relevant.

It should also recognise that applications increasingly operate across multiple networks. The correct developer abstraction may eventually be closer to "Give this application enough trustworthy connectivity to perform this task at this location, time, cost and security level". This should be technology-aware but also technology-neutral, perhaps involving 4G, 5G NSA, 5G SA, Wi-Fi, Private 5G, satellite D2D etc in different contexts.

That is a more useful long-term abstraction than making every developer understand mobile-network architecture.

---

5. How can adoption be accelerated?

The industry should probably expose **fewer APIs more successfully**, rather than continuously increasing the API count. More fanciful notions such as edge computing APIs should be de-prioritised. Immediate priorities should include:

- genuinely useful capabilities
- broad multi-operator availability including small MNOs and MVNOs
- predictable semantics
- excellent documentation
- realistic sandboxes
- simple progression from test to production
- transparent pricing
- operational support
- clear consent handling
- reliable capability-discovery mechanisms.
- regulatory clarity, for example around what privacy, Open Internet and sovereignty rules mean for APIs and their back-end platforms.

Developers should also be able to determine whether an API is useful before paying to invoke a control function. For example, which network is the device currently using, what performance is currently available, can the application requirements already be met, is QoD available at this location and for this subscription and what confidence does the operator have that the requested result can be delivered?

This makes informational and discovery APIs an important prerequisite for advanced programmability.

---

6. Main drivers of successful adoption

The most important driver is not standardisation in isolation. It is:
ubiquity + consistency + discoverability + economic usefulness + proof points

An API that works perfectly for 15% of an application's users may have less value than an inferior mechanism working for 95%. Successful network APIs therefore need sufficient coverage across both operators and access environments.

They also need semantic consistency. A QoS profile carrying the same label on two networks does not necessarily produce the same latency, throughput, congestion behaviour, geographic availability or indoor experience. Applications therefore need capability discovery and graceful degradation.
The application needs to know not merely that an API exists, but **whether its requested capability is available here and now for this device over this path**.

---

7. Main technical and commercial challenges

7.1 Multi-SIM and eSIM identity

Multi-SIM handling is already recognised inside CAMARA.
The current QoD specification warns that where multiple devices are associated with the same phone number, using that phone number alone may be insufficient to identify which device should receive enhanced QoS. Depending on implementation, the API might reject the request, apply it to all devices in the group or apply it to an unintended device.

But that is only one form of multi-SIM complexity. A growing number of users may have:

- a home physical SIM plus travel eSIM
- two personal subscriptions
- personal plus enterprise eSIM
- separate voice and data subscriptions
- smartwatch/companion SIMs
- data-only eSIMs.

It is quite possible that in a few years, the average user may have 2+ active eSIMs in their smartphone, and some may have 3, 4 or more. This can affect:

- Number Verification
- SIM Swap
- network-based silent authentication
- roaming status
- QoD
- location
- device reachability.

A SIM Swap query against the user's “base” number remains meaningful even while the application uses another data connection. By contrast, a network-based authentication mechanism that implicitly assumes that the connection carrying the application traffic corresponds to the phone number being authenticated may encounter ambiguity.

Enterprise eSIMs add another issue. Legitimate changes in enterprise-managed profiles or devices may resemble security-relevant SIM events unless the semantics are sufficiently precise.

BEREC should therefore encourage APIs to distinguish explicitly between:

- device identity
- subscription identity
- phone number/MSISDN
- current access subscription
- subscription about which information is requested.

"The user", "the device" and "the phone number" should not be treated as interchangeable concepts.

7.2 Granular knowledge of coverage and capacity

QoD, QoS booking, slice booking and edge APIs depend on much more accurate network knowledge than is usually acknowledged. QoD cannot create radio coverage where none exists. It cannot necessarily produce additional capacity in a severely constrained cell. And it cannot either divert a request to Wi-Fi if that has higher availability, or indeed diagnose or fix an unrelated Wi-Fi bottleneck.

Booking future QoS or a network slice is more demanding still. CAMARA's QoS Booking API requires parameters including a QoS profile, start time, duration and service area. Network Slice Booking similarly reserves network resources within designated areas and time periods.

This implies a substantial hidden dependency on:

- coverage prediction
- capacity forecasting
- cell-load prediction
- spectrum configuration
- planned network changes
- indoor penetration
- neutral-host arrangements
- user mobility
- admission control.

For many enterprise applications, building-level granularity matters. A nominal outdoor 5G coverage map is not enough to determine whether a camera works inside basement level 2, a robot remains connected behind industrial machinery, an augmented-reality application works on floor 34 or if the 5th live broadcaster can obtain sufficient uplink capacity from inside a stadium.

Predictive Connectivity Data is an interesting step because CAMARA explicitly models expected future coverage using antenna and cell-load information. However, the same concept needs to extend to **serviceability confidence**. An advanced API should ideally be able to communicate:

- capability available
- capability unavailable
- capability uncertain
- expected performance range
- confidence level
- geographic scope
- time validity.

A binary API success code is not enough.

7.3 Indoor and neutral-host boundaries

Indoor wireless is a particularly important blind spot. Many high-value enterprise applications occur in factories, hospitals, airports, offices, stadiums, shopping centres, warehouses, stations etc.

Those are precisely the locations where the logical mobile operator may not fully control the physical infrastructure. A DAS or neutral-host system may carry the MNO's customers while using:

- shared radios
- venue-owned infrastructure
- third-party fibre
- local edge systems.

This creates an architectural question. Who is responsible for fulfilling the API? The MNO may authenticate the subscriber and expose the API while the infrastructure determining actual radio performance is operated by another party. In some circumstances, the neutral host may operate its own core network (using "MOCN" sharing).

BEREC should therefore investigate whether advanced network APIs need mechanisms for **federated fulfilment**, where an MNO-facing API can obtain capabilities from neutral hosts, private networks or managed indoor systems. A similar situation may apply in roaming situations.

Without this, an API may work perfectly outdoors and quietly stop being meaningful when the user walks into a building.

7.4 Consent and delegated authority

Consent is considerably more complicated than simply adding an OAuth flow. CAMARA's Consent Info API already acknowledges this by allowing applications to determine whether processing is permitted for a particular user, scope and purpose, and where necessary obtain an operator-managed URL through which consent can be captured. It also explicitly supports backend applications where the user is not actively interacting with a device.

The important policy questions include:

- Is permission granted for one API call?
- One session?
- One application?
- One purpose?
- One enterprise?
- A fixed period?
- Until explicitly revoked?
- Does materially changing the purpose require re-authorisation?
- Can an AI agent inherit or delegate authority?
- Can an enterprise administrator grant permission for an entire fleet?
- How is the consent interaction surfaced inside a mobile app rather than a browser?

The answers will differ by API and legal basis. Location deserves particular attention because it can be highly sensitive and because repeated calls reveal substantially more than a single location query.

But requiring an interactive consent screen for every API call would make many legitimate services unusable. At the opposite extreme, indefinite blanket consent for any future use of location data would be difficult to justify.

BEREC should also distinguish **legal basis** from the narrower concept of user consent. Explicit consent will not necessarily be the appropriate legal basis for every anti-fraud or enterprise-processing scenario.

Consent issue will be further complicated by the use of AI agents either on behalf of the user, or the enterprise / app developer. Can an AI consent to revealing its / its owner's location? If a user tells their phone's AI agent to "*pay the restaurant bill*" or "*get my pharmaceutical prescription picked up and delivered to my house*", what happens next?

IoT devices

IoT exposes the limitations of smartphone-oriented consent models. A sensor, camera, robot or utility meter may have no screen, no browser and no human operator at the moment an API is invoked. In industrial IoT, there may not even be a human data subject associated with the connectivity event.

In other cases, such as connected vehicles, wearable devices or employee equipment, there may be – but is consent needed for everyone involved, such as a vehicle's passengers as well as drivers? A physical device should not be assumed to be capable of clicking "Allow".

7.5 VPNs, privacy tools and security overlays

VPNs and enterprise security mechanisms create another underexplored boundary. A VPN does **not** necessarily prevent the mobile network from knowing which SIM is connected, which cell serves it or whether the subscription is roaming. But it can obscure the application's actual destination, application-specific traffic flows and the relationship between source IP and application identity.

This could break various network APIs, such as an edge-discovery mechanism or various QoS / QoD tools, especially if multiple applications are aggregated inside an encrypted tunnel. The network may see the VPN flow rather than the application flows inside it.

This scenario is increasingly relevant because security architectures (especially for enterprise users and business-critical systems) are moving towards always-on VPNs, zero-trust access, SASE, privacy relays, encrypted DNS and application-layer tunnels.

Network APIs should therefore be explicit about what they can and cannot infer when security overlays are present.

7.6 AI agents as network API consumers

The aforementioned January 2026 **Unthinkable Lab: Agentic AI + Telecoms APIs = ?**, which I co-led with Andrew Collinson, examined this issue with more than 30 participants from operators, vendors, API organisations, developers, regulators and other stakeholders.

The subsequent report concluded that agentic AI, supported by MCP and related mechanisms, could become a significant driver of telco API consumption, but also emphasised that the market is currently extremely immature.

The February follow-up discussion with Raman Mistry of GSMA Open Gateway reinforced several important findings. First, actual use of MCP plus agentic AI plus Open Gateway APIs is currently largely at proof-of-concept stage. Second, APIs are **components rather than complete products**. Channel partners, software providers and vertical specialists are likely to assemble them into outcomes.

Importantly, an AI agent can reason across several possible ways of satisfying any given intent. For example, tasking an agent to "**Improve the quality of this video upload"** might request QoD, query Connectivity Insights, move to Wi-Fi, change codec, reduce frame rate, switch to another eSIM, tether through another device or postpone the upload. A similar set of issues might occur if an agent seeks to determine whether a banking transaction might be fraudulent.

It is unclear whether this will lead to overall greater use of APIs, or more competition (and thus lower pricing) of the various alternatives.

In the TelecomTV discussion I described this as a potential **bypass problem**. If network APIs are difficult to discover, expensive or incomplete, agents may simply select another route to the requested outcome. This makes agentic AI fundamentally different from conventional API consumption, as an ordinary application is usually hard-coded to invoke a particular API, while an agent can decide **not to use it**.

There should also be a strong distinction between "**read-only APIs"** such as checking current network conditions, vs. "**state-changing or chargeable APIs"** such as purchasing QoD, booking a slice or changing network policy.

MCP can provide a useful tool-discovery and invocation layer. CAMARA itself is now examining this model and describes MCP as a bridge through which AI applications can discover and call network capabilities. But MCP is **not the authorisation, consent or policy layer**. A generic AI agent should not receive unrestricted network-control credentials simply because it supports MCP.

Relevant Unthinkable and related material

- Unthinkable Lab Report: Accelerating Telco APIs with Agentic AI
- TelecomTV webinar/panel: Can agentic AI unlock the telco API opportunity?
- Unthinkable Lab Analyst's Overview Slides
- My review: CAMARA position paper on MCP
- My summary of the Unthinkable Lab findings
- Andrew Collinson's post on the Lab report
- Andrew Collinson on why network APIs matter for agentic AI
- Connective Insight

These discussions reinforce an important point for BEREC: agentic AI may increase API consumption, but it will also make APIs compete dynamically against alternative technical solutions.

8. Regulatory challenges

Several existing regulatory areas clearly intersect with network APIs, including GDPR, ePrivacy, security legislation, competition rules and the Open Internet Regulation. There may also be touch-points with new rules on sovereignty.

For example, for QoD / QoS type APIs, BEREC should assess the **underlying network behaviour**, rather than the fact that an API is involved. Exposing a traffic-management function through an API does not inherently make that function either compliant or non-compliant with Open Internet requirements. The relevant issue is what differentiation occurs for whom / which application, for what purpose, for how long and with what effect on other users.

Similarly, identity or location APIs should be assessed according to the underlying data, purpose, legal basis and authority rather than placed into a generic network-API regulatory category.

Overall, I would caution against broad API-specific regulation at this stage. The market remains immature and architecture, distribution and commercial models are unsettled.

However, BEREC should instead actively monitor:

- discriminatory access
- exclusion of MVNOs
- exclusive aggregation arrangements
- excessive switching costs
- concentration among intermediaries
- tying APIs to other operator or cloud products
- lack of commercial portability
- differential treatment of competing applications
- unclear responsibility across neutral-host and private-network boundaries.

There is also a European sovereignty dimension. If European network capabilities ultimately reach developers only through one or two global cloud or aggregation platforms, European operators may retain physical infrastructure while losing much of the innovation and commercial layer. On the other hand, if API security – or the security of an operator's core network - is improved by using non-European software elements, there is clearly a trade-off to consider.

---

9. How interoperable are current APIs?

Not answered

Conclusions and recommendations

Network APIs are potentially useful building blocks for future communications services, but expectations should remain proportionate to evidence. The telecom industry has a long history of assuming that exposing network capabilities will automatically create developer ecosystems.

Standardisation removes an important barrier. But it does not remove the need for:

- compelling applications
- broad coverage of networks, including in rural areas and indoors
- useful economics
- good developer experience
- accurate network information
- reliable cross-network operation.

BEREC should also recognise a deeper architectural change. The eventual consumer may no longer be a developer writing static code against an operator API. It may be an AI agent deciding dynamically how to satisfy an intent, which *might* conclude that the best answer is to buy QoD via an API, but which also may conclude that the best answer is simply to use Wi-Fi. That possibility needs to be built into both regulatory thinking and market forecasts.

I recommend that BEREC:

1. distinguish clearly between API specification, deployment, production use and commercial success
2. request quantitative evidence of paid production usage and revenue
3. distinguish identity/fraud, information, communications and network-programmability API markets
4. distinguish between internal and external consumption of network APIs, and between "raw" APIs and API-enabled products and solutions
5. broaden its analytical model beyond public mobile networks to include Wi-Fi, fixed networks, private networks, neutral hosts, indoor systems and satellite connectivity
6. explicitly examine transitions among 4G, 5G NSA and 5G SA rather than treating "5G" as a homogeneous capability
7. require clearer semantics around device, subscription and phone-number identity for multi-SIM and eSIM users
8. examine travel eSIM, enterprise eSIM, companion SIM and data-only subscription scenarios in API testing
9. recognise granular coverage, capacity and serviceability information as a prerequisite for many QoD, slice and edge APIs
10. investigate indoor and neutral-host API fulfilment and responsibility
11. evaluate consent by purpose, duration, application, delegation and revocation rather than treating it as a single binary state
12. develop explicit models for IoT and fleet authorisation where no interactive end-user interface exists
13. examine the effect of VPNs, SASE, privacy relays and other security overlays on API validity and network visibility
14. assess how voice, messaging and network APIs will be combined commercially through CPaaS and other platforms

15. recognise vertical solution providers such as fraud and identity specialists as a distinct part of the value chain
16. analyse AI agents as first-class API consumers, including tool discovery, delegated authority, auditability, rate limits and spending controls
17. distinguish read-only information APIs from state-changing and chargeable network-control APIs in agentic environments
18. model substitution between APIs rather than adding individual theoretical TAMs together
19. explicitly model non-API substitutes such as Wi-Fi, application adaptation, alternative SIMs, local compute and alternative identity signals
20. monitor concentration and switching barriers among aggregators and channel providers
21. avoid premature mandates specifying individual APIs or commercial models
22. commission empirical cross-network and cross-operator interoperability testing
23. focus regulation on actual network behaviour and market outcomes rather than the presence of an API interface.

The best future for network APIs is probably not one in which millions of developers become experts in mobile networks. It is one in which applications, platforms and agents can request communications capabilities when they genuinely need them, while understanding their price, limitations and alternatives. CAMARA and Open Gateway can form an important part of that architecture, but they should not be mistaken for the architecture as a whole.