

Network APIs, Explained.

- 1 What's driving Network API adoption?
- 2 What can you build with Network APIs?
- 3 Who's using Network APIs in production?
- 4 Why do most Network API projects stall?
- 5 What does a working implementation look like?
- 6 What's next for Network APIs?



Your phone connects to a cell tower. The mobile network operator knows your number, your approximate location, whether your SIM changed recently, how congested the local network is, and whether your device is even reachable.

For years, that data stayed locked inside telecom infrastructure. Operators ran on proprietary protocols that required specialized expertise to access. No REST endpoints. No standardization across carriers. Little/No incentive for operators to expose these capabilities to developers.

That is now changing.

Telecom operators are now exposing this data through standardized REST APIs. The same network intelligence that powers operator systems is accessible to developers building fraud detection, identity verification, asset tracking, and bandwidth-sensitive applications.

The trigger was pain.

SIM swap fraud surged 1,055% in the UK last year.⁵ T-Mobile paid \$33 million to settle a single attack in March 2025.⁶ Banks started asking operators for help. Logistics companies wanted location data that works indoors. Broadcasters needed guaranteed bandwidth. The demand got loud enough that operators finally got aligned on a standardized interface.

This white paper breaks down why Network APIs are gaining traction, what you can build with them, and how to ship without getting stuck in integration limbo.



1 What's Driving Network API Adoption?

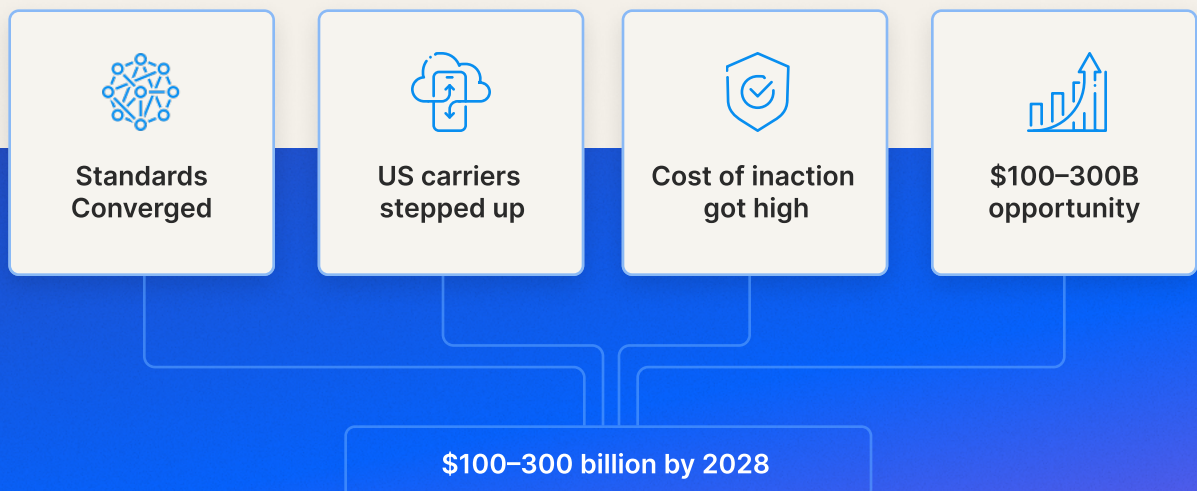
Network APIs have existed for years. Here's what shifted:

Standards finally converged. CAMARA Project, an open-source initiative by the Linux Foundation is defining standardized telecom APIs for developers. The initiative now has 60 standardized APIs with 1,300+ contributors from 476 organizations. Nokia, Ericsson, Deutsche Telekom, Orange, Vodafone, and 30+ other telecom operators and vendors are aligned on the same specs.¹ For the first time, developers can target a common API model instead of building custom integrations for every operator.

US carriers stepped up. In 2025, AT&T, T-Mobile, and Verizon deployed SIM swap and number verification APIs. First time all three aligned on network API offerings.²

The cost of inaction got too high. Fraud lawsuits. Regulatory pressure. Enterprise customers demanding capabilities operators were sitting on. Cloud-native architecture made APIs practical to expose. McKinsey now projects \$100–300 billion in connectivity and edge-computing revenue over the next five to seven years.³ Ericsson puts the direct API opportunity at \$20 billion by 2028.⁴

The window opened. The question now is who moves fast enough to use it.



2 What Can You Build With Network APIs?

Start with the problems. Find solutions.

SMS verification is vulnerable. Attackers intercept OTPs. SIM swaps redirect codes. Network-based Number Verification sidesteps all of it: the operator confirms the number matches the device making the request. No SMS. Nothing to intercept.

Account takeovers start with SIM swaps. Attacker ports your number to their SIM. Intercepts your 2FA codes. Drains your account.

SIM Swap Detection flags recent SIM changes before you authorize anything sensitive. UK incidents jumped from 289 to nearly 3,000 in 2024.⁵

GPS is useless indoors. Network-based Device Location uses cell tower triangulation. Works in buildings, underground, in urban canyons.

Banks cross-verify transaction locations. Logistics companies track assets without relying on device sensors which are battery-intensive.

Live video can't buffer. Quality on Demand (QoD) allows temporary bandwidth prioritization for specific sessions. At MWC Barcelona 2025, Nokia demonstrated a car in Finland driven remotely from the show floor, with video feed guaranteed by API.⁷ RAI hit 60 Mbps uplink for live broadcasts using network slicing.⁸

IoT commands fail silently. Device Reachability checks if a device is connected to the network before you send commands. Simple, but critical when failed messages mean failed operations.

And that's just the starting list. Ride-sharing apps verifying driver location matches the pickup point. Emergency services locating callers when GPS is unavailable. Event ticketing confirming a buyer is physically at the venue for will-call. Field service teams checking if a technician's device is reachable before dispatching.

The capabilities are real and tangible. Shipping them is the hard part.

3 Who's Using Network APIs In Production?

The use cases span industries.

A luxury retailer could verify that a high-value online purchase is happening from the same location as the customer's device, not from a fraudster halfway across the world. A content creator going live to a million followers could lock in guaranteed bandwidth before hitting "Start": no buffering, no dropped frames.

These aren't hypotheticals. The building blocks exist. And some companies are already in production:

Itaú. Brazil's largest private bank, runs over 5 million SIM swap validations monthly. Every sensitive transaction checks whether the customer's SIM changed recently. Account takeover attempts get flagged before they complete.⁹

Daycoval deployed the same API during digital channel expansion. Identity theft attempts now get caught at registration, not after the damage is done.¹⁰

frete.com, a Brazilian freight company, verifies driver identity through Know Your Customer-Match API. Network-verified onboarding, not just document uploads.¹¹

RAI guarantees bandwidth for live broadcasts. 60 Mbps uplink during events, with priority access when networks are most congested.⁸

These aren't experiments. They're production systems handling millions of transactions.



4 Why Do Most Network API Projects Stall?

The APIs are standardized. Integration still isn't.

Authentication flows don't work like typical OAuth. Error handling requires understanding operator-specific quirks. You need UI components (verification flows, location displays, bandwidth controls) that don't exist in your component library.

Here's the real problem: an API that returns coordinates isn't a map. An API that checks SIM swaps isn't a fraud prevention flow. You still have to build the experience layer. That takes weeks. Most teams don't have weeks.

The gap between "API access" and a "shipped feature" is where projects die.

One way to close that gap: don't build the experience layer from scratch. Use pre-built components (prefabs for single functions, journeys for multi-step workflows) that come with the API integration, UI, and error handling already done.

5 What Does A Working Implementation Look Like?

Let's make this concrete. Say you want to add SIM swap detection to your banking app.

Without pre-built components: You authenticate with the operator's API (different from your usual OAuth). You handle the request/response cycle. You build error handling for operator-specific edge cases. You design the UI: what happens when a SIM swap is detected? Block the transaction? Step-up authentication? You build that flow. You test across operators. You maintain it. That's months of work before you ship anything.

With a tool like WaveXD: WaveXD is a component marketplace built on Nokia's Network-as-Code platform. You grab the Secure Auth prefab. It handles number verification, SIM swap detection, and device location in one component.



The API integration is done. The UI is done. You configure it, drop it in, ship.

If you need a multi-step flow (SIM check plus location verification plus step-up auth), you use a Journey. Anti-Fraud chains those checks together. Asset Tracking combines location with QoD for fleet monitoring.

Want to see how it all fits together?

[WaveBank](#) is a complete neo-banking reference app.

[TrackMate](#) shows logistics tracking end-to-end.

Deployment is flexible. Export as standard Web Components that work in Angular, React, Vue, vanilla JS. No platform lock-in. Or build inside WaveMaker Studio for faster iteration with drag-drop assembly.

[WaveMaker Studio](#) is the broader platform these components plug into. Two things matter here:

Two-Pass Architecture. Most AI code generation hallucinates. WaveMaker splits the process: Pass 1 uses AI to generate a structured meta model (layout, components, design tokens, guardrails). Pass 2 runs a deterministic transpiler, with no AI, that outputs production-grade Angular, React, or React Native.

Same input, same output, every time.

Add to this AI Agents. Specialized AI agents in Wavemaker studio that handle UI generation, API binding, security config, architecture scaffolding. Every output is previewable, governable, reversible. You stay in control.

The math: weeks of telecom-specific integration becomes hours. You ship network-powered features without becoming a telecom expert.

6 What's Next For Network APIs?

CAMARA's API catalog keeps expanding. Network slicing (dedicated virtual networks for specific applications) is moving from demos to commercial deployment.

The use cases are widening. Gaming platforms selling priority connectivity as a paid feature. Insurance companies verifying location claims through network data. Healthcare apps guaranteeing bandwidth for telemedicine.

A year ago, adding SIM swap detection meant carrier partnerships and months of integration. Today, it's a component.

The gap between what's possible and what's practical is closing fast.

If you're building anything that touches identity, location, or connectivity, this is worth paying attention to.

Citations

- 1 [CAMARA Project](#). "Fall 2025 Meta-Release."
- 2 [Fierce Network](#). "What's next for 5G network APIs?" December 2025.
- 3 [McKinsey & Company](#). "What it will take for telcos to unlock value from network APIs." February 2024.
- 4 [NETSCOUT](#). "APIs: The Gateway to Dynamic 5G Networks." March 2025.
- 5 [Cifas](#). "Fraudscape 2025: Reported fraud hits record levels." April 2025.
- 6 [Greenberg Glusker](#). "Secures Landmark \$33M Arbitration Award Against T-Mobile." March 2025.
- 7 [Fierce Network](#). "MWC: I drove a car in Finland - remotely from Barcelona." March 2025.
- 8 [Commsbrief](#). "How Network APIs Could Unlock \$300 Billion in 5G Revenue." 2025.
- 9 [Telefónica Open Gateway](#). "Itaú: More secure transactions implementing Open Gateway."
- 10 [Telefónica Open Gateway](#). "Finance & insurance services."
- 11 [Telefónica](#). "David del Val shares Open Gateway milestones." January 2025.



WaveXD is a marketplace of ready-to-use prefab components that simplify the use of 5G-driven features in a variety of applications.

Discover pre-wired components, journeys, apps, and themes that you can embed, compose, customize, and extend.



Visit the [WaveXD Marketplace](#).



WaveXD is developed and maintained by [WaveMaker](#), a leading agentic app development platform, **trusted by Fortune 500 companies.**