AvidThink

Sponsored by amdocs

Effective Telco API Monetization Strategies

Five Guiding Principles for Communication Service Providers

RESEARCH BRIEF

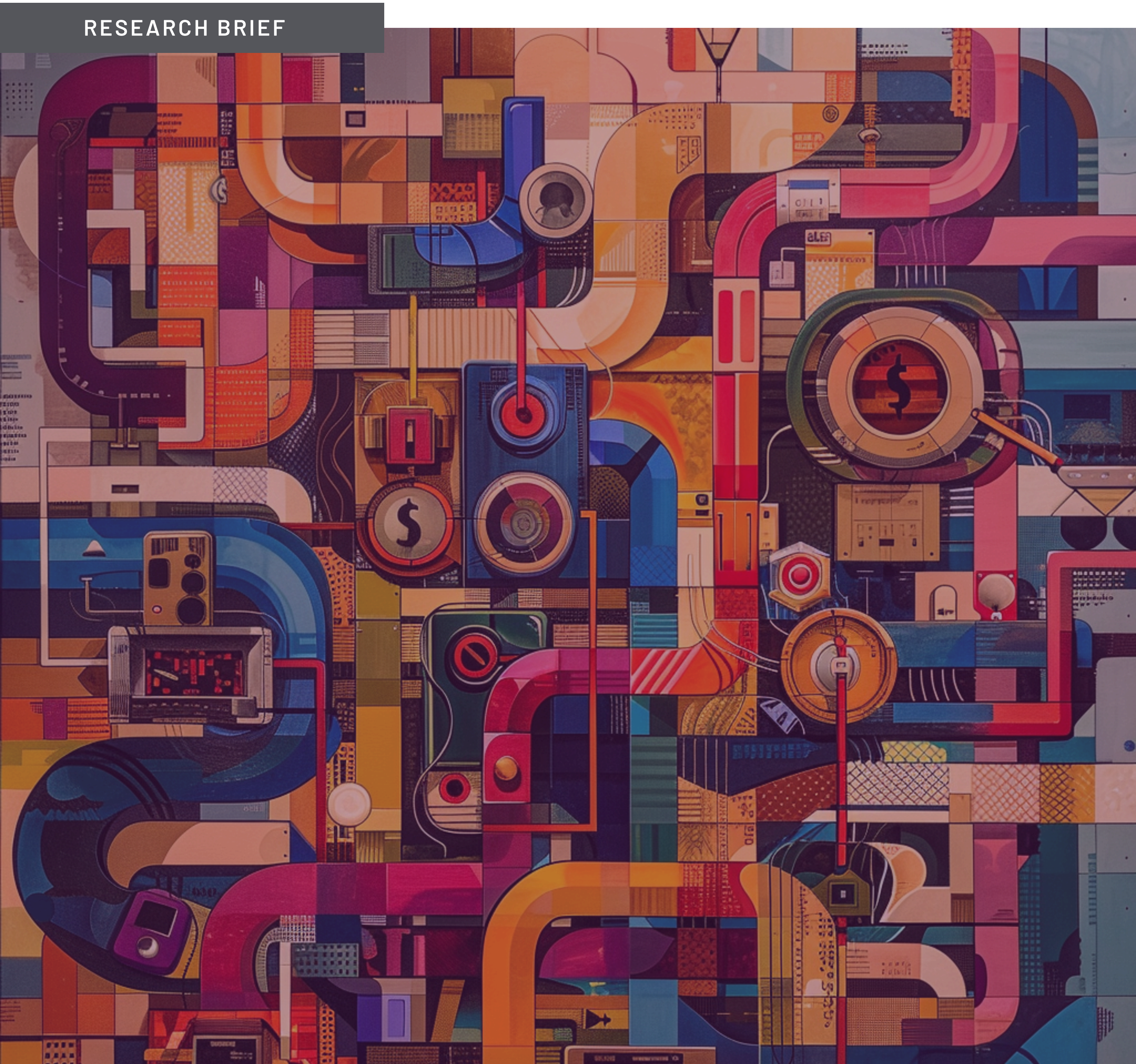


Table of Contents

Research briefs are independent content created by analysts working for AvidThink LLC. These reports are made possible through the sponsorship of our commercial supporters. Sponsors do not have any editorial control over the report content, and the views represented herein are solely those of AvidThink LLC. For more information about report sponsorship, please reach out to us at research@avidthink.com.

About AvidThink

AvidThink is a research and analysis firm focused on providing cutting-edge insights into the latest in infrastructure technologies. Formerly SDxCentral's research group, AvidThink launched as an independent company in October 2018. AvidThink's coverage includes 5G infrastructure, private wireless, edge computing, SD-WAN, SASE, SSE, ZTNA, cloud and containers, SDN, NFV, and infrastructure security. Our clients range from Fortune 500 enterprises and hyperscalers to tier-1 communications service providers, fast-growing unicorns, and innovative startups. AvidThink's research has been quoted by Forbes, Wall Street Journal, Light Reading, Fierce Telecom, Fierce Wireless, Mobile World Live, and other major publications. Visit AvidThink at avidthink.com.

RESEARCH BRIEF

Effective Telco API Monetization Strategies

Five Guiding Principles for Communication Service Providers

Executive Summary

Communication Service Providers (CSPs) seek new revenue streams by leveraging emerging technologies like programmable networks, APIs, AI/ML, and Generative AI. With the recent buzz around telco APIs, the GSMA Open Gateway initiative, and the related Linux Foundation CAMARA Project, telco APIs are viewed as the next vehicle to enable the monetization of operator networks and services.

However, succeeding in API monetization requires CSPs to lay the appropriate groundwork across culture, processes, and platforms. Based on conversations and learnings from successful API strategies of cloud providers and CSPs at different maturity levels, this paper presents five fundamental principles to guide CSPs on their API monetization journey. It highlights the vital role of end-to-end service orchestration (E2ESO) and key capabilities like dynamic inventory, assurance, and agile charging in enabling effective API monetization.

Introduction

The telecommunications industry is undergoing a profound transformation. With traditional connectivity becoming increasingly commoditized, Communication Service Providers (CSPs) are urgently exploring new avenues for revenue growth across enterprise and consumer segments. Network monetization has been a key theme at major telecom conferences like Mobile World Congress over the past few years. Still, the challenge isn't limited to 5G mobile networks—it's across all telco services.

Emerging technologies like programmable networks, artificial intelligence/machine learning (AI/ML), generative AI, APIs, and network exposure capabilities generate significant interest as potential monetization opportunities. By opening up their networks and exposing capabilities via APIs, CSPs aim to enable a broad range of innovative use cases across industries, from improved fraud detection in financial services to optimized video streaming and gaming services. This network exposure is intended to help lay the groundwork for more bandwidth-intensive and latency-stringent coming down the line, including immersive metaverse and extended reality (XR) experiences.

However, as they pursue these opportunities, CSPs must remain grounded and recognize the importance of laying a solid foundation and adhering to proven best practices. Many technologies are still nascent, and monetization models need to be proven. CSPs need to remain agile and focus on capability building at the core while remaining flexible on the configuration of these services. This is akin to creating the right Lego brick components that can be stacked together flexibly to form different structures — what those structures are, and which are the most popular remain an area of exploration.

Having engaged with CSPs at different stages of their API monetization journeys and observed the successful API-centric strategies of cloud giants like Amazon, Google, and Microsoft, we've distilled five fundamental principles that can help guide CSPs as they embark on their network API initiatives — the goal is to unlock value at every stage of the API journey while staying on track towards the end goal. These principles warrant careful consideration for any telco executive or strategist grappling with questions around APIs, programmability, exposure, and monetization.

Five Guiding Principles for Effective API Monetization

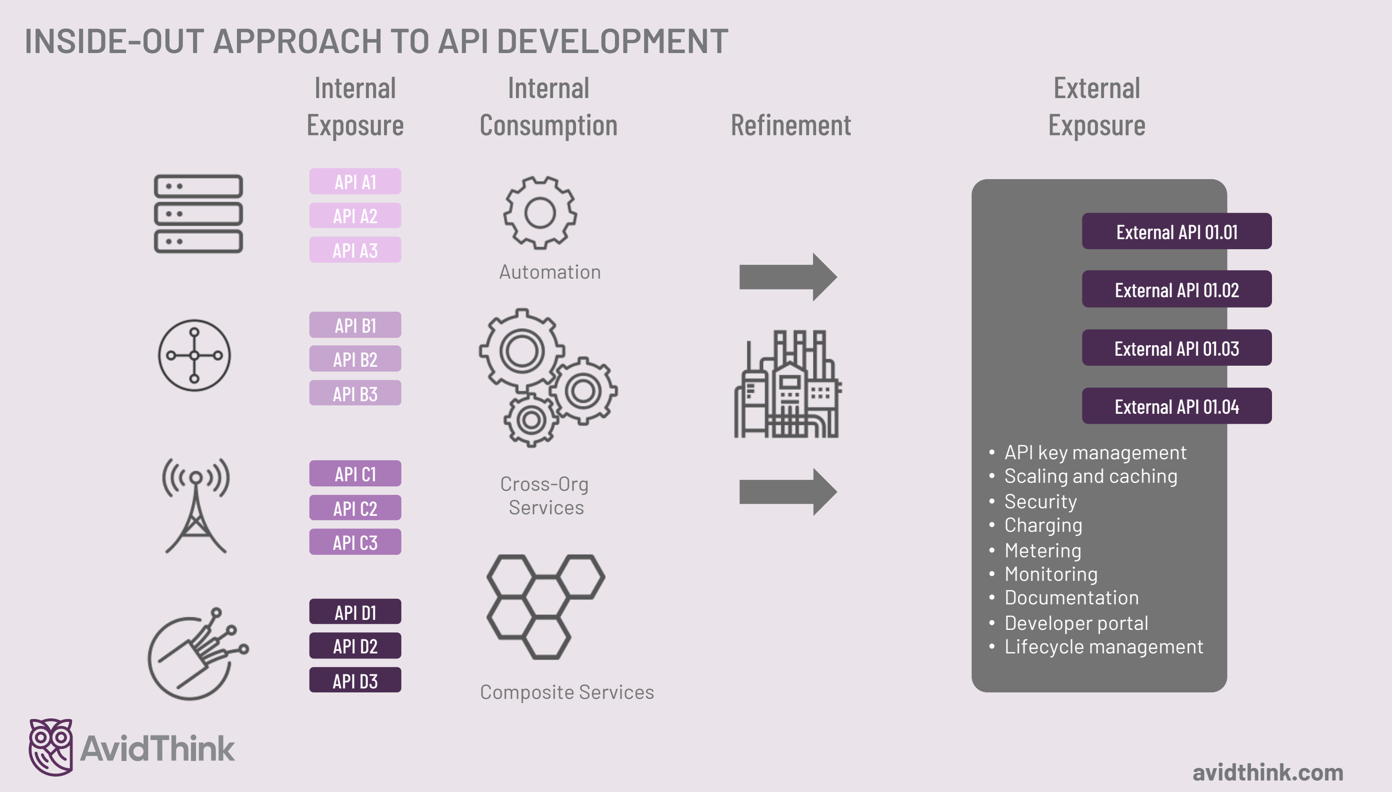
1. Embrace API-centricity as a culture, not a one-off project

CSPs' first and most important principle is to approach API enablement as a cultural and organizational transformation, not a finite destination to reach. Becoming an API-centric business requires rethinking how networks, systems, and services are designed, deployed, and operated with APIs at the core, not as an afterthought.

We advise CSPs to take a gradual, measured approach to API productization, learning from the evolutionary journeys and best practices of hyperscalers and web-scale companies. Amazon's transformation is particularly instructive. Back in 2002, then-CEO Jeff Bezos issued his now-famous mandate that all teams must expose their data and functionality through APIs, that all inter-team communications had to be exclusively through those APIs (no exceptions), and that those APIs should be designed from the ground up to be externalizable. Notably, though, not all APIs were immediately exposed externally.

In the early years, Amazon focused on its internal APIs. This allowed the company to build robust API development capabilities, establish replicable design practices, and fine-tune APIs based on internal feedback before eventually opening them to external consumption. Like many startups that advocate internal use of their own products to drive rapid improvements, Amazon forced itself to "eat its dog food " (or "drink its own champagne") and become its first and best customer for its APIs.

This progressive, iterative, "inside-out" approach to API development — harnessing APIs to drive agility and efficiency internally before selectively exposing them externally — has been employed to significant effect by many successful internet-era companies. Internal APIs have and should be used to enable automation, consistent internal architectural frameworks, rapid prototyping and evolution with loose coupling between organizational units, and reduce the spaghetti-like mess for internal system projects. For CSPs, it highlights the importance of patience and deliberation. Internal learning and capability building are critical before large-scale commercialization.



As CSPs build confidence, they can carefully expose a limited set of APIs externally to partners and developers, focusing on those that drive quick impact through targeted use cases. For instance, the fraud detection APIs currently being developed within the GSMA CAMARA initiative are likely a safe and valuable starting point for many operators. Close monitoring of early rollouts is essential to gather feedback and refine the API product design, business model, and supporting processes before expanding further.

2. Focus API design on service enablement; standards like CAMARA are a means to an end

The second fundamental principle is that not all APIs are equal in business value. The most impactful APIs will enable CSPs to deliver and monetize new services, not generic capabilities. CSPs must approach API design from an outside-in service enablement mindset.

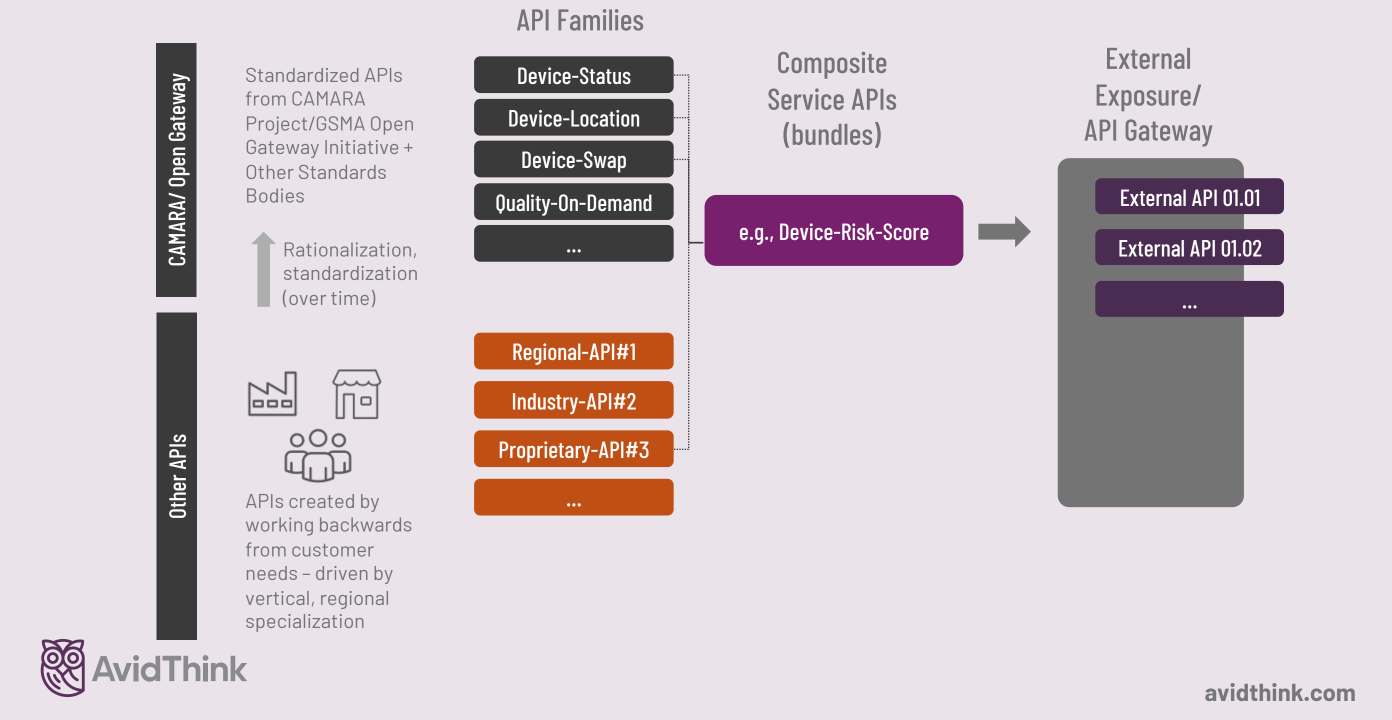
CSPs should begin by understanding the services that will be exposed through APIs and the value that they provide to enterprise and consumer customers across different verticals and use cases. Working backwards allow CSPs to determine what functionality to expose, what configurability and management to offer, and therefore, what APIs to design.

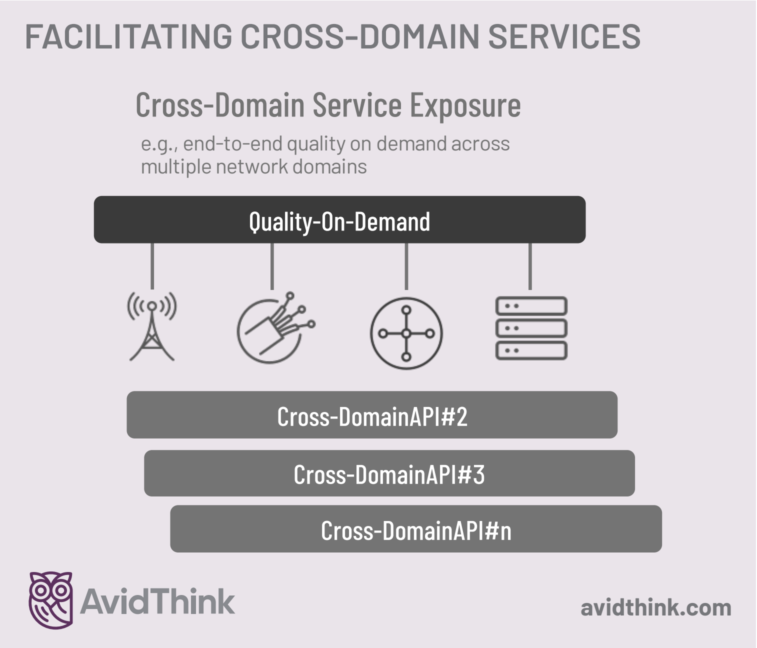
In this context, it's essential to recognize that while industry-level API standardization initiatives like CAMARA are incredibly valuable to drive developer adoption, CSPs shouldn't view them as the end goal. GSMA Open Gateway initiative and CAMARA project provide a standard baseline set of API definitions to build, but CSPs must be prepared to go beyond.

To differentiate, CSPs may offer better quality, higher performance, or more cost-effective services via these standard APIs than their competition. They can also group together a broader range of standards-based APIs for a more comprehensive set of services — for example, starting with basic fraud checks (checking if a SIM was swapped or performing account verification), expanding to layering in location information, and then adding quality-on-demand capabilities. Leading CSPs can create new services and bring them to open source and standards bodies for ratification and normalization, helping advance the telco API movement.

Beyond improved service quality, or bundling and unbundling, we recommend that CSPs carefully consider the creation of customized, operator-specific, or regional versions and extensions of standard APIs to enable differentiation, especially in highly competitive markets. Vertical industry needs and unique regional requirements can drive CSPs to extend and create additional API families. Looking at past protocols and standards like RADIUS for authentication or SNMP for device management, we see how the industry evolved from standards-only to support proprietary extensions with vendor-specific attributes and management information bases (MIBs). We acknowledge the risk of standards fragmentation but as long as CSPs support CAMARA standards as the baseline, the industry should hold together.

EXTENDING BEYOND CAMARA — ADDITIONAL APIS AND COMPOSITE SERVICES


API Families
CAMARA/ Open Gateway
Standardized APIs from CAMARA Project/GSMA Open Gateway Initiative + Other Standards Bodies
Rationalization, standardization (over time)
Other APIs
APIs created by working backwards from customer needs – driven by vertical, regional specialization
Device-Status
Device-Location
Device-Swap
Quality-On-Demand
...
Regional-API#1
Industry-API#2
Proprietary-API#3
...
Composite Service APIs (bundles)
e.g., Device-Risk-Score
External Exposure/ API Gateway
External API 01.01
External API 01.02
...
AvidThink
avidthink.com


FACILITATING CROSS-DOMAIN SERVICES
Cross-Domain Service Exposure
e.g., end-to-end quality on demand across multiple network domains
Quality-On-Demand
Cross-DomainAPI#2
Cross-DomainAPI#3
Cross-DomainAPI#n
AvidThink
avidthink.com

Nevertheless, one critical advantage CSPs have in the API economy is their unique ability to stitch together and expose cross-domain service capabilities that span their entire infrastructure from network edge to core. Narrow APIs limited to a single network domain like the RAN or 5G core have their uses, but the CSP's market power lies in integrating multiple underlying capabilities into high-value, end-to-end service APIs.

Consider quality-on-demand (QoD) as an example. An API that allows application providers to dynamically request and configure a particular assured network performance within the 5G core is functional. But an API that enables end-to-end QoD from an end user's device to the application's endpoint, crossing the RAN, transport, and core networks and into the data center, is far more valuable. The same applies to other lucrative areas like SASE (Secure Access Service Edge), where combining networking and security capabilities across domains into an integrated API is crucial to the value proposition — a single API call that can provision a network path with specific security attributes could be both convenient and valuable.

To enable composite service APIs, CSPs need a robust cross-domain orchestration and exposure layer to abstract the underlying complexity and provide a unified, intent-based interface to the application and content provider partners. With it, CSPs can avoid getting mired in low-level, fragmented APIs that fail to realize the full business potential.

As we wrap up this principle, we remind CSPs to let target use cases and business models drive the API design, not vice versa.

3. E2E service orchestration is a key pillar in API monetization

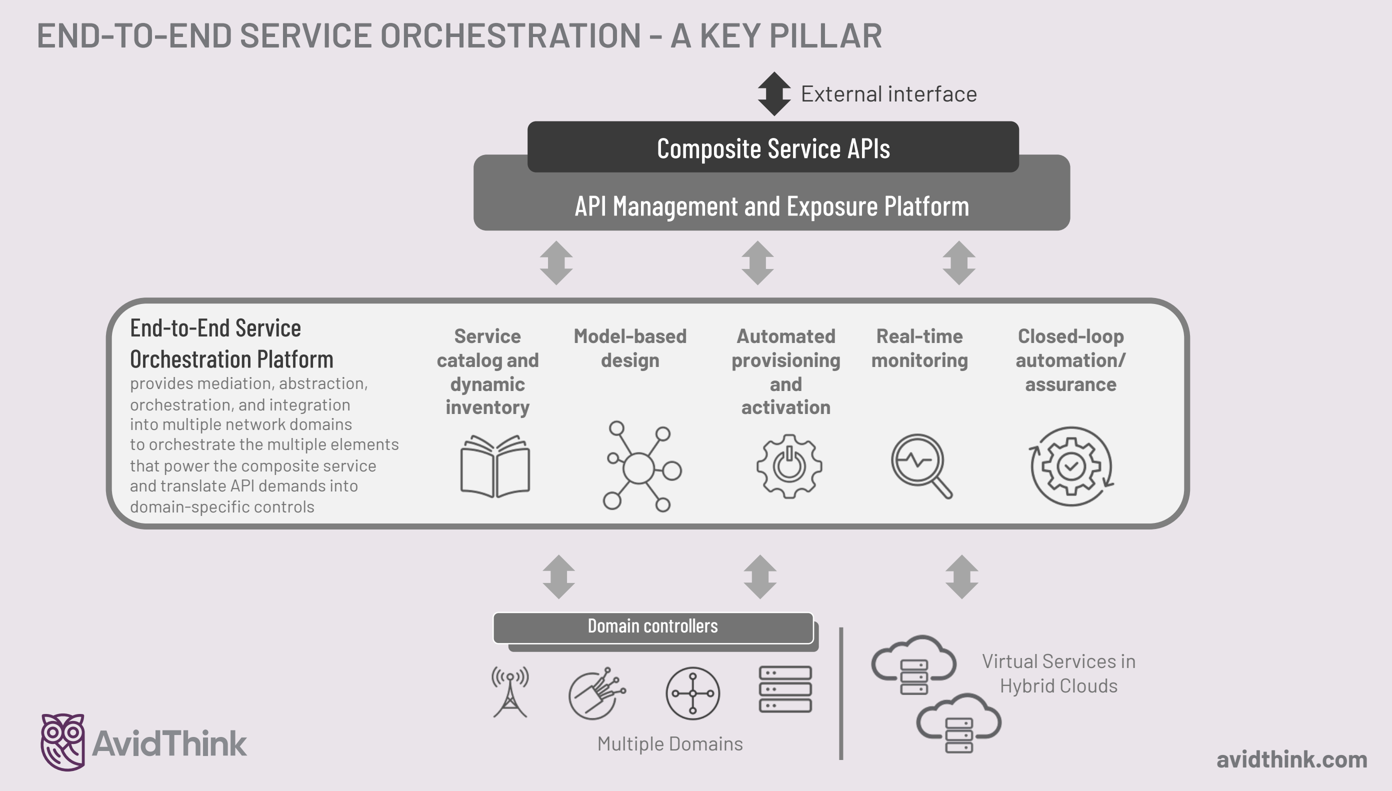
The third principle is that effective productization and monetization of APIs require CSPs to be operationally ready with the proper underlying platforms and processes for end-to-end management.

Exposing an external API is only the tip of the spear. Below the surface, careful coordination between multiple underlying networks and systems is required, as well as the ability to dynamically orchestrate and assure the performance of the services to which the APIs are linked. In fact, for composite APIs that invoke multiple services, CSPs need to be able to seamlessly stitch together and manage capabilities across different technology and operational domains — from partner- and customer-facing portals through the BSS/OSS layer, down to the network — and potentially even across operator and partner (hyperscaler, inter-exchange centers, data center) boundaries.

This is where having a robust, standards-based E2ESO (end-to-end service orchestration) becomes pivotal. An advanced E2ESO layer needs to be supported by vital operational capabilities including:

- Unified, cross-domain service catalog and dynamic inventory.
- Model-driven service design and fulfillment.
- Automated service provisioning and activation across both physical and virtual infrastructure.
- Continuous, real-time service monitoring and assurance.
- Closed-loop automation driven by AI/ML.

E2ESO is the linchpin to rapidly designing, deploying, and monetizing API products at scale while meeting demanding SLAs. Without this ability to take an end-to-end view of services driven by APIs and automate their lifecycle, CSPs will struggle to operationalize their API portfolios in an agile, cost-effective manner. They will likely be forced to limit themselves to exposing simple, siloed capabilities rather than integrated cross-domain offers.


END-TO-END SERVICE ORCHESTRATION - A KEY PILLAR
External interface
Composite Service APIs
API Management and Exposure Platform
End-to-End Service Orchestration Platform
provides mediation, abstraction, orchestration, and integration into multiple network domains to orchestrate the multiple elements that power the composite service and translate API demands into domain-specific controls
Service catalog and dynamic inventory
Model-based design
Automated provisioning and activation
Real-time monitoring
Closed-loop automation/ assurance
Domain controllers
Multiple Domains
Virtual Services in Hybrid Clouds
AvidThink
avidthink.com

4. Treat monetization and API product management as a first-class citizen

The fourth principle is to recognize that APIs are a means to an end —enabling new business models and revenue streams. We learned painful lessons from early 5G rollouts, where CSPs invested in building advanced networks without clear monetization plans. Unfortunately, monetization has become the last step in the process when it should have initially been treated as a first-class citizen. CSPs can ill afford to make the same mistake with their upcoming API programs.

With APIs, the key is to take an outside-in approach led by the business and product teams rather than the network and engineering teams. Start by engaging with enterprise customers, application developers, and partners to understand their needs, then map them to API-enabled service offerings and business models. Market test and validate that the target customers are willing to pay for the APIs and associated SLAs before finalizing the product design and underlying technical architecture.

It's a complete reversal of the "build it and they will come" approach that has characterized many prior telco initiatives. The business case and go-to-market plan should drive the technology roadmap, and tight collaboration between business and network teams is essential — taking more of a "buy it and we will build it" mindset. The good news is that APIs inherently allow for much more rapid prototyping, iterating, and trialing of offers compared to traditional telco services.

In tandem, CSPs must evolve their monetization platforms to support the unique aspects of productizing APIs. Legacy telco charging systems designed for basic connectivity and CDRs (call detail records) are ill-suited to the demands of API business models. API offerings may involve nuanced usage-based pricing, time-of-day considerations, tiering, revenue sharing, limits and throttling, zero-rating, subscriptions, and even third-party settlements.

The table stakes for API charging are modern, cloud-native monetization platforms that provide API providers and consumers with precise, real-time visibility into pricing, usage, account balances, and other financial data . These systems should be tightly coupled with API gateways to enable seamless end-to-end processes like customer onboarding, usage tracking, and invoice generation. Enterprise customers are already accustomed to fine-grained and near-real-time billing and charging dashboards from hyperscalers and other web-based platforms. They will expect the same from telco API services.

Additionally, having a flexible, configurable monetization layer creates opportunities for CSPs to differentiate their API products in a crowded market, even if the underlying APIs are similar to competitors. Creative bundling and unbundling of APIs, QoS tiers, and consumption models can unlock new value propositions.

5. Obsess over API product usability and developer experience

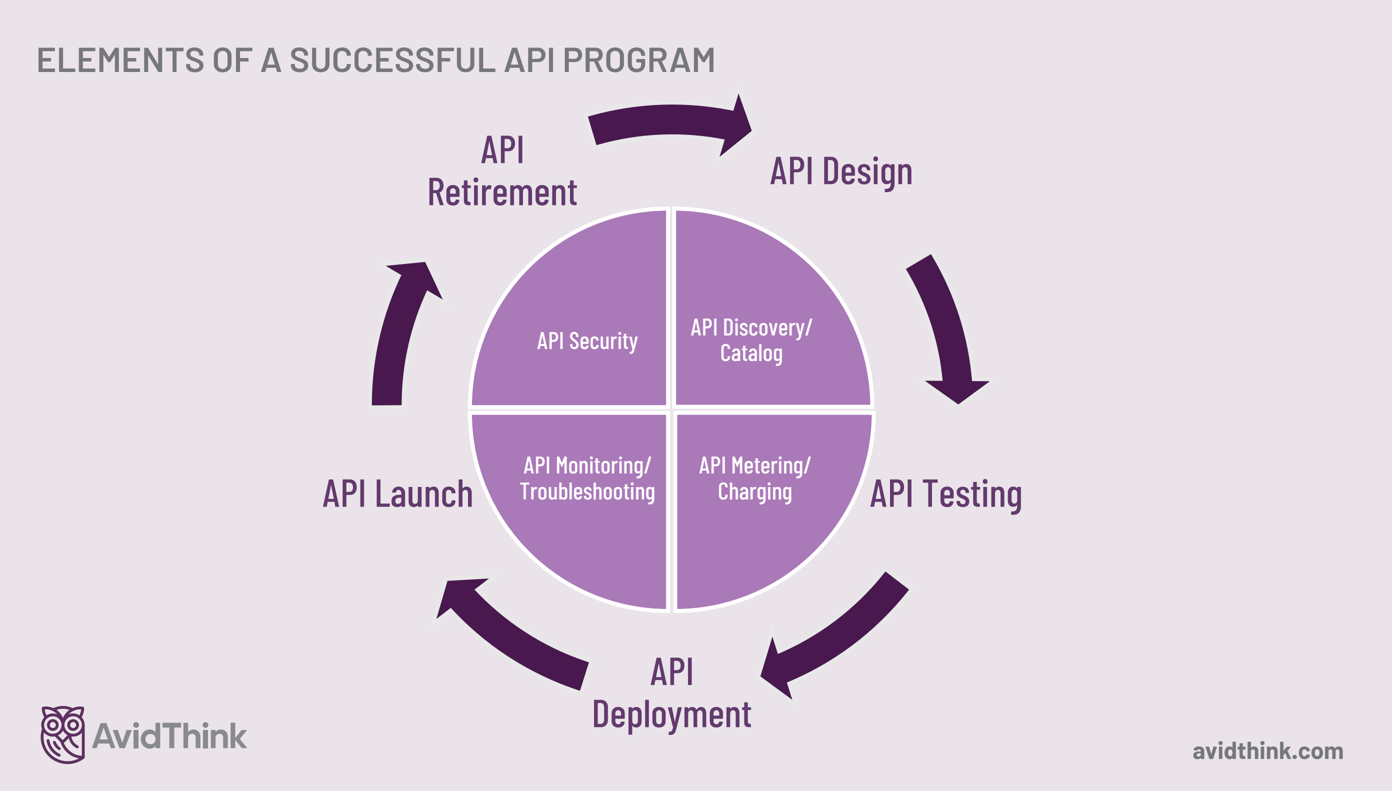
The fifth and final principle for CSPs is to realize that the success of any API program depends on adoption and usage by developers. Unlike traditional telco-managed services, APIs are meant to be building blocks for innovation by customers and partners. As CSPs embark on API productization, they must focus on two key aspects driving consumption: API product usability and developer experience.

On the usability front, CSPs should strive to design APIs that are intuitive, self-explanatory, and easy to integrate for developers who may have limited telco domain expertise. It is critical to hide unnecessary complexity and expose APIs in language familiar to mainstream developers.

In addition to the APIs, CSPs must offer appealing developer portals and tools for API discovery, sandbox playgrounds, interactive documentation, account management, support, and performance monitoring. The developer experience needs to be on par with what developers are accustomed to from the leading cloud platforms and API providers.

This is a considerable shift for operators used to building services for a handful of large, well-known enterprise, wholesale, and MVNO customers with dedicated account teams and bespoke solutions. Operators who do this in-house may need to hire and empower dedicated developer relations teams to proactively engage and support large numbers of diverse API consumers. However, as the CAMARA and GSMA frameworks point out, there are partners in the ecosystem, such as the hyperscalers, who can work closely with operators on developer recruitment, onboarding, and management. Hyperscalers already have a great relationship with developers and can play an aggregation role. CSPs will, however, need to consider that this level of disintermediation by hyperscalers, while adding value, can reduce the CSPs' market power .

Beyond supporting common standards from CAMARA and best practices from carrier-focused organizations like TM Forum, CSPs must invest in end-to-end lifecycle product management for API as a product consumed by developers at partners and enterprises. A successful API program requires management across conception, launch, and retirement of the service:



- API design and development: Adhering to API design best practices such as RESTful architecture, consistent versioning, clear documentation, and backward compatibility.
- API testing and deployment: Ensuring APIs are rigorously tested and can be rolled out in a fully automated manner across distributed cloud environments via CI/CD pipelines.
- API launch: Defining the API product, SLAs, pricing/business model, and launch plan through CSP or aggregator marketplace.
- API retirement: Having clear policies and processes for deprecating and retiring APIs that are no longer needed without impacting customers.

And underlying the API lifecycle are key supporting capabilities required in the API platform, including:

- API discovery: Provide the means for developers to browse a catalog or find the appropriate APIs for services they are trying to incorporate in their application.
- API security and access control: Putting in place robust security controls, authentication, role-based access controls, and throttling to protect the APIs and underlying systems.
- API monitoring and troubleshooting: Proactively exposing and monitoring API and associated network service uptime, performance, and error rates and quickly identifying and resolving any issues .
- API metering and charging: Accounting for API usage at an appropriate level of granularity and enabling near-real-time updates with flexible tagging (attribution) to allow charging analysis by customers.

To excel in these areas, CSPs must adapt their operating models and invest in API management technologies and modern, cloud-native OSS/BSS capabilities to handle automated fulfillment and assurance of dynamic services driven by API calls. Many CSPs will find it valuable to partner with leading ISVs who can provide pre-integrated, best-of-breed API lifecycle management solutions.

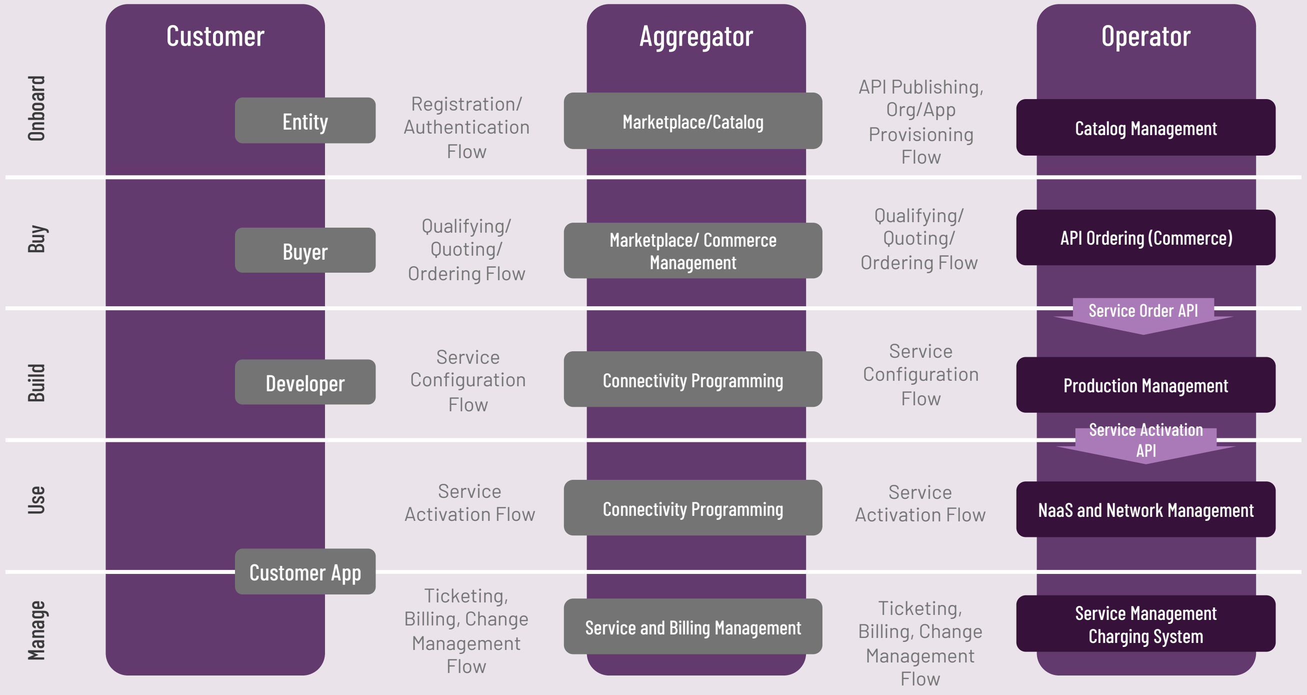
An Architecture for API Monetization

Pulling together the key themes from the principles above, we can start to envision what a reference architecture for effective API monetization might look like for a CSP. At a high level, it requires several essential layers:

- A unified API exposure layer that provides a common entry point for all API requests and handles essential cross-cutting functions like security, traffic management, metering, and developer portal.
- An E2ESO layer that takes the API requests from the exposure layer and intelligently decomposes and orchestrates them into the appropriate domain-level actions across the infrastructure. The E2ESO layer can also provide intent-based interfaces (as those become defined) that drive the underlying actions using lower-level APIs.
- Domain-specific orchestrators and controllers (e.g., for 5G core, transport, edge) that translate the high-level service intent from the E2ESO layer into granular configuration and activation of network functions and elements. This is where much of the heavy lifting of service provisioning and assurance happens.
- Unified OSS/BSS capabilities like service catalog, inventory, order management, flexible charging, policy management, monitoring, and closed-loop assurance, fed by cross-domain network and service data, to enable end-to-end management of the API-driven services. Critically, all these capabilities must be built on a standard, cloud-native software foundation for seamless interworking.

An excellent proof point of how an architecture like this can be made real to enable new API services is the TM Forum Catalyst project SHINE (Simple Hyperscaler Integrated Network Experience), which brought together CSPs and partners to demonstrate cloud-to-network integration via a unified API exposure and management layer.

TM FORUM CATALYST SIMPLE HYPERSCALER INTEGRATED NETWORK EXPERIENCE (SHINE)


Customer
Aggregator
Operator
Onboard
Entity
Registration/ Authentication Flow
Marketplace/Catalog
API Publishing, Org/App Provisioning Flow
Catalog Management
Buy
Buyer
Qualifying/ Quoting/ Ordering Flow
Marketplace/ Commerce Management
Qualifying/ Quoting/ Ordering Flow
API Ordering (Commerce)
Service Order API
Build
Developer
Service Configuration Flow
Connectivity Programming
Service Configuration Flow
Production Management
Service Activation API
Use
Service Activation Flow
Connectivity Programming
Service Activation Flow
NaaS and Network Management
Customer App
Manage
Ticketing, Billing, Change Management Flow
Service and Billing Management
Ticketing, Billing, Change Management Flow
Service Management Charging System

AvidThink

Source: TM Forum (abstracted by AvidThink)

avidthink.com

Conclusion

There is little doubt that APIs will be central to the next wave of telco business model innovation and growth. As 5G and cloud-native networks mature, CSPs have an immense opportunity to create and monetize a rich portfolio of API products. However, capturing this opportunity will require CSPs to think and operate in fundamentally new ways.

Rather than the traditional model of building discrete network assets and seeking to find use cases, value creation and extraction in the API economy requires an outside-in, market-driven approach to innovation and monetization. CSPs can maximize their chances of success in this fast-moving space by starting with a clear view of target customers and their unmet needs and then working backward to design API products and business models to serve them.

Even as GSMA and the CAMARA project normalize, expand, and evolve standard-based APIs, CSPs should build their API capabilities in parallel. CSPs must now invest to participate in the new API ecosystem and capture learnings early.

AvidThink, LLC
1900 Camden Ave
San Jose, California 95124 USA
avidthink.com

Rev A