



www.pipelinepub.com

Volume 22, Issue 10

From Automation to Orchestration: Redefining Operational Agility

By: [Ionut Grosu](#)

We Modernized the Enterprise. Why Is Change Still Slow?



Communications Service Providers (CSPs) have modernized OSS and BSS platforms, moved workloads to the cloud, exposed APIs, automated manual processes, and invested heavily in digital customer experiences. Those programs have delivered real gains in efficiency, scalability, and service delivery.

Yet a familiar contradiction remains. Launching a new partner, commercializing a digital service, or supporting a new pricing model can still take months and require coordinated changes across multiple systems and teams.

The bottleneck has moved. Most transformation programs optimized the enterprise while the environment around it became far more complex. CSPs now operate within networks of hyperscalers, software vendors, cybersecurity providers, distributors, managed service providers, systems integrators, and AI platforms. Each participant brings its own products, APIs, lifecycle events, pricing structures, billing models, operational processes, and commercial obligations.

Operational agility is therefore no longer determined solely by how efficiently an organization runs its

own systems. It increasingly depends on how effectively it coordinates the broader ecosystem in which it operates. The unit of optimization is shifting from the enterprise to the ecosystem.

Research from McKinsey & Company points in the same direction: the next wave of telecommunications growth depends less on isolated technology modernization and more on transforming operating models through digital technologies, AI, automation, and ecosystem collaboration.

The Architecture Problem Is Not a Shortage of APIs

The telecommunications industry has spent decades integrating systems, and integration remains essential. But integration alone does not create agility.

An API makes a capability accessible; it does not automatically make that capability independent, reusable, or safe to compose. A collection of APIs wrapped around tightly coupled processes can simply create a more accessible monolith. If launching a product still requires synchronized changes across five systems, three teams, and two partners, the architecture is not composable regardless of how many APIs it exposes.

Integration answers, “Can these systems communicate?” Orchestration answers, “How does the business outcome progress across them?” True composability requires capabilities with clear boundaries, stable contracts, independent lifecycles, and business meaning. Orchestration then coordinates those capabilities across an end-to-end journey, maintaining state, enforcing dependencies, handling exceptions, and ensuring that each participant completes its part.

An ordering API, a pricing service, an entitlement capability, and a billing service are useful individually. They create a business outcome only when the order can move reliably from qualification and pricing through provisioning, activation, billing, support, renewal, and partner settlement. End-to-end process logic should not be buried in point-to-point integrations or owned by the channel that initiated the transaction.

Composable Capabilities Need a Common Commercial Language

Heterogeneous providers rarely describe the same commercial event in the same way. One provider may treat a service as a subscription, another as a metered entitlement, and another as a license with its own rules for activation, suspension, renewal, and cancellation. Passing those differences directly into every downstream process merely distributes vendor-specific complexity across the architecture.

CSPs need a thin normalization and abstraction layer: canonical commercial concepts for products, offers, orders, subscriptions, entitlements, usage, billing, and settlement, mapped to each provider at the edge. The objective is not to erase legitimate provider differences. It is to stop every channel and process from having to understand them independently.

This is where TM Forum’s Open Digital Architecture provides a useful industry direction: modular, interoperable components and standardized Open APIs that can be assembled and evolved more independently. Standards reduce unnecessary variation, but the architectural discipline still matters. Components must own coherent capabilities; APIs must express durable contracts; and orchestration must remain separate from both channels and systems of record.

The Product Catalog Is Part of the Architecture, Not the Center of It

Product catalogs must evolve beyond passive repositories, but they should not absorb ordering,

provisioning, billing, and lifecycle execution. That would replace one monolith with another.

The catalog should be part of a composable commercial architecture. It supplies product definitions, relationships, rules, and metadata that reusable pricing, qualification, ordering, fulfillment, entitlement, billing, and partner capabilities consume. Channels use those shared capabilities; they do not recreate the business logic inside a marketplace, portal, CRM workflow, or reseller application.

The architectural objective should be simple: integrate once and commercialize many times. Once a provider or product is connected, the same operational capabilities should support direct enterprise sales, an SMB marketplace, a reseller channel, a managed-services bundle, an API channel, and AI-assisted commerce, without six separate integrations.

What This Looks Like: Onboarding a Cybersecurity Partner

Consider a CSP adding a managed endpoint-security service. In a traditional model, the provider's product and APIs trigger separate changes in the CRM, catalog, marketplace, ordering stack, provisioning workflow, billing platform, support tools, partner reporting, and settlement process. Channel-specific logic grows around the integration. A change to the provider's licensing or renewal model then propagates across the estate.

In a composable model, the provider is mapped once into normalized product, order, entitlement, usage, billing, and lifecycle capabilities. An orchestration policy defines the journey: validate customer eligibility, price the offer, secure approval, place the partner order, confirm activation, create entitlements, begin billing, monitor exceptions, and allocate partner revenue. The marketplace, account team, reseller portal, and API channel invoke the same capabilities.

The launch still requires product, commercial, security, and operational work. Composability does not eliminate that work; it prevents the same work from being rebuilt for every channel and bundle. When the provider introduces a new tier, the CSP changes the relevant product and orchestration policies rather than redesigning the entire transaction path.

Partner Ecosystems Require Network Thinking

Traditional telecommunications value chains were relatively linear: vendors supplied, distributors distributed, resellers sold, and customers consumed. Digital ecosystems behave more like networks. A CSP may act as a marketplace operator, service provider, reseller, distributor, technology partner, and ecosystem orchestrator, sometimes within the same customer relationship.

Products can pass through multiple commercial layers before reaching the customer, while entitlement, billing, revenue sharing, settlement, support, and lifecycle management span several organizations. Treating every relationship as an individual integration project does not scale.

The better model is a reusable set of partner capabilities: onboarding, identity and authorization, catalog federation, commercial agreements, ordering, provisioning, billing, revenue allocation, settlement, reporting, and lifecycle management. Composable architecture makes those capabilities reusable. Orchestration makes them operational.

A Marketplace Is a Channel, Not the Operating Model

Digital marketplaces matter, but the storefront is only the customer-facing layer. Behind a single transaction sit discovery, configuration, eligibility, pricing, contracting, ordering, provisioning, usage collection, billing, settlement, entitlements, renewals, upgrades, and support.

Customers experience one transaction. The organization coordinates dozens of operational interactions. A marketplace layered over inflexible OSS/BSS processes does not create agility; it gives customers a modern interface to an inflexible operating model.

The competitive advantage lies beneath the marketplace. When channels consume shared commercial capabilities, a CSP can introduce another marketplace, partner portal, assisted-sales journey, or API route without rebuilding the operational foundation.

Agentic AI Will Expose Architectural Weaknesses

AI can forecast demand, recommend products, optimize pricing, identify anomalies, and automate customer interactions. As agentic systems mature, they will also initiate and coordinate more complex commercial and operational workflows.

An AI agent does not need another user interface. It needs reliable, governed, executable capabilities it can call. An agent may understand a request to move 500 employees to a new collaboration suite, add endpoint security, preserve the current budget, and activate everything next month. Executing that intent still requires product discovery, eligibility, pricing, approval, ordering, provisioning, entitlement, billing, and settlement across multiple providers.

AI determines what should happen; composable architecture and orchestration determine how it happens. Agentic commerce will expose architectural weaknesses quickly because an agent can automate only what the enterprise has made reliably executable. If capabilities have ambiguous ownership, inconsistent data, hidden dependencies, or manual exception paths, AI will accelerate confusion rather than outcomes.

CSPs that establish reusable capabilities, canonical commercial concepts, machine-readable policies, observability, and controlled orchestration today will be far better positioned to operationalize AI tomorrow.

Measure Ecosystem Agility, Not Only System Performance

Availability, transaction speed, automation rates, and unit cost remain important, but they do not reveal whether the ecosystem can change efficiently. Leaders should also measure the marginal effort and cost of ecosystem expansion.

In practice, that means tracking a handful of ecosystem-level measures: how long it takes to onboard a new provider, from technical kickoff to a production-ready connection, and how long from commercial agreement to the first billable customer event; how quickly a multi-provider bundle can be launched; and what share of customer journeys reuse common pricing, ordering, entitlement, and billing capabilities rather than channel-specific logic.

Two trends reveal whether the operating model actually scales. Integration effort per new partner should decline as the ecosystem matures, and the incremental operating cost of adding a provider, product, channel, or commercial model should keep falling. A rising volume of manual interventions and exceptions, meanwhile, is an early signal that complexity is being absorbed by people rather than by the architecture.

Behind all of these sits one question: does operational complexity increase every time the ecosystem grows?

Build for Ecosystem Elasticity

The purpose of composability is not modularity for its own sake. It is ecosystem elasticity: the ability to add partners, products, channels, and commercial models without operational complexity growing at the same rate.

That principle changes architecture decisions. New channels should consume capabilities rather than own process logic. Provider differences should be normalized at the edge rather than propagated through the enterprise. Orchestration should coordinate outcomes without hard-coding them into integrations. Product definitions should inform execution without turning the catalog into a new center of gravity. And every new participant should strengthen a reusable operating model rather than create another bespoke path.

The communications industry will only become more interconnected. CSPs will collaborate with hyperscalers, software vendors, AI platforms, distributors, and emerging technology partners while expanding digital services, consumption models, and routes to market. Supporting that future with another custom integration whenever the ecosystem changes is neither efficient nor sustainable.

The organizations that succeed will not necessarily have the largest networks, the broadest portfolios, or the most sophisticated storefronts. They will have something more valuable: an operating model designed to adapt.

The defining question for operational agility is no longer, “How quickly can our organization change?” It is, “How quickly can our ecosystem change with us?”