

1 + 1 + 1 > 3

The Architecture of a Composable Broadband Business

Building OSS/BSS That
Works Better Together

By David Tidd
VP of Engineering
ETI Software

integrate™

Contents

1. The Platform Should Be the Multiplier
2. The Problem with Single-Sourced Architecture
3. Single Source of Truth $\neq$ Single Source of Technology
4. Standardize the Foundation. Not the Business.
5. The Architecture of Composition
6. From Connected Systems to Composable Capabilities
7. The Common Model: Creating Shared Context
8. Replaceability Is an Architectural Advantage
9. From Data to Closed-Loop Operations
10. The Platform Belongs to the Business
11. What This Means for Broadband Providers
12. The Last Word: Your Business Is the Product
13. About ETI Software Solutions



1. The Platform Should Be the Multiplier

"The whole is greater than the sum of its parts." - Aristotle

Modern broadband providers already operate complex technology ecosystems. Network infrastructure, OSS/BSS, CRM, billing, GIS, provisioning, telemetry, workflow, security, analytics, AI, and other capabilities may come from many different vendors.

The goal should not be to eliminate that diversity. It should be to make that diversity work together.

The real value comes from composition:

Independent best-of-breed capabilities + open interfaces + orchestration + shared context = multiplicative value.

Each capability may be valuable independently.

But when those capabilities can share information, trigger processes, participate in common workflows, and operate within a shared business context, something different happens.

$$1 + 1 + 1 > 3$$

That is platform value.



2. The Problem with Single-Sourced Architecture

There is an understandable appeal to buying everything from one vendor. Fewer suppliers. One technology stack. One ecosystem. One roadmap. But simplicity at the point of purchase can create constraints later.

A giant vendor-controlled platform tends to optimize every component around its own ecosystem. That can make it harder to introduce a better technology, preserve an existing investment, integrate an acquisition, support a different network vendor, or build a capability that differentiates your business.

Broadband providers don't stand still.

Networks evolve. Vendors change. New technologies emerge. Customer expectations shift. Acquisitions introduce different systems. AI creates entirely new requirements for access to operational data.

An architecture designed around a single supplier can eventually make change more difficult precisely when the business needs greater flexibility.

**A single vendor can provide a product.
A platform should enable an ecosystem.**



3. Single Source of Truth ≠ Single Source of Technology

These two ideas are easy to confuse.

A business may absolutely need authoritative sources for customer, service, inventory, network, billing, and operational information. But establishing authoritative data does not mean one technology supplier needs to own every system that creates, consumes, or acts on that information.

That distinction matters.

Single source of truth

Creates consistency around the information the business uses to operate.

Single source of technology

Creates dependency on one supplier to provide the capabilities the business uses to operate.

The first can increase clarity. The second can reduce choice.

A modern architecture should allow providers to establish trusted business and data models while maintaining the freedom to select the right technologies for individual functions.

You want consistency without dependency.



4. Standardize the Foundation. Not the Business.

Every service provider is different.

Different customers, markets, network technologies, vendor relationships, geographic footprints, operational processes, products, pricing models, business rules, and competitive advantages.

Those differences are not problems for a platform to eliminate. They are the reason the platform exists.

The platform should standardize how capabilities connect without standardizing what makes the business unique.

That creates a powerful architectural principle:

Standardize the foundation.

Compose the capabilities.

Differentiate the business.

Each can contribute its strengths to a larger operational system.



5. The Architecture of Composition

A composable broadband architecture is not simply a collection of disconnected products. The pieces need connective tissue.

That foundation can include:

Open standards

TM Forum, OpenAPI, AsyncAPI, YANG, NETCONF/gNMI, and other standards create consistent ways for technologies to communicate.

Best-of-breed components

Technologies such as Kubernetes, Kafka, Keycloak, Prometheus, Grafana, OpenTelemetry, and others can provide specialized capabilities without requiring the entire architecture to come from one supplier.

Vendor neutrality

Network and technology providers can participate through adapters and standardized interfaces.

A common canonical model

A shared model creates a semantic contract so every system does not have to independently define what the data means.

An integration fabric

The integration layer connects systems, data, events, and workflows so independently useful technologies become composable business capabilities.

Shared observability and policy

Telemetry, security, identity, audit, and governance can span the ecosystem instead of remaining isolated inside individual applications.

Replaceability

Individual components can evolve or be replaced without requiring the entire architecture to be redesigned.

Together, these principles create something much more powerful than conventional point-to-point integration. They create a platform designed for change.



6. From Connected Systems to Composable Capabilities

Connecting two applications is useful. But connectivity alone is not the objective. The business process is.

Consider a customer order.

That single action might need information or actions from CRM, billing, inventory, provisioning, GIS, network systems, device platforms, workflow tools, and third-party services.

Traditional integration tends to ask:

How do we connect System A to System B?

Composable architecture asks:

What needs to happen when the customer places the order?

That shift changes the role of integration.

Instead of merely moving information between applications, the integration layer helps turn individual systems into reusable business capabilities.

Customer qualification, provisioning, device activation, service assurance, and customer notification become capabilities.

Those capabilities can then be orchestrated differently depending on the service, customer, network, or business requirement.

**The value isn't the connection.
The value is what the connection makes possible.**



7. The Common Model: Creating Shared Context

Connecting systems solves only part of the problem. Systems also need to understand one another.

One platform may describe a customer, service, location, device, or network resource differently from another. Without a shared context, every new connection requires another translation.

Over time, that creates an increasingly complicated web of mappings and dependencies, but a common canonical model changes the equation.

Instead of every system defining the meaning of the data for every other system, the platform provides a shared semantic contract. Individual applications remain independent, and their data becomes understandable across the ecosystem.

This is especially important as automation and AI become part of broadband operations. Intelligent systems need more than access to data. They need context.

**Connected data tells you what happened. Context helps
you understand what it means.
A shared model creates the foundation for that
understanding.**



8. Replaceability Is an Architectural Advantage

Technology changes. Your architecture should expect it.

A better network platform may emerge. A provider may change billing systems. An acquisition may introduce an entirely different OSS/BSS environment. A new AI capability may need access to data that existing applications were never designed to expose.

In a tightly coupled architecture, replacing one component can affect everything around it.

In a composable architecture, the interface matters more than the implementation behind it. That means individual components can be upgraded, replaced, or expanded while the broader operational model remains intact.

Replaceability is not an admission that a technology will fail. It is recognition that the business will evolve. The architecture should make that evolution easier.



9. From Data to Closed-Loop Operations

The greatest opportunity appears when individual technologies stop functioning as isolated tools and begin participating in a shared operational loop.

Imagine these capabilities working together:

Network controller + telemetry + inventory + service catalog + workflow + AI + observability

Connected through a composable platform, they can support an entirely different operating model:



The value of that closed loop is greater than the value of any individual technology used to create it.

10. The Platform Belongs to the Business

Vendor neutrality is about more than procurement flexibility. It changes who owns the value created by the architecture.

In a closed ecosystem, the supplier determines which technologies participate, how they interact, and how quickly the ecosystem evolves.

In an open, composable architecture, the business can decide.

- Keep an existing investment that still performs well.
- Introduce a better capability when one becomes available.
- Build proprietary technology where differentiation matters.
- Integrate new network vendors.
- Connect acquired systems.
- Experiment with AI without rebuilding the operational stack.

The objective is not to own every component. The objective is to make every component more valuable when connected to the others.

And when the architecture is open, that multiplier belongs to the business rather than to a single technology vendor.



11. What This Means for Broadband Providers

The question for broadband operators is no longer simply:
Which platform should we buy?

A better set of questions is:

- Can we **integrate the technologies we already have?**
- Can we **choose the best capability for each function?**
- Can our **systems exchange information using open interfaces?**
- Can we **create shared context across operational and business systems?**
- Can **business processes span multiple applications and vendors?**
- Can we **introduce automation without rebuilding everything underneath it?**
- Can we **replace individual components without redesigning the entire architecture?**
- Can our **data support analytics and AI across organizational silos?**
- Can we **create capabilities that reflect how our business operates?**
- Does the **architecture increase our freedom as technology changes?**

The answers reveal something important: architecture is not merely a technology decision.

Architecture determines how easily the business can change.

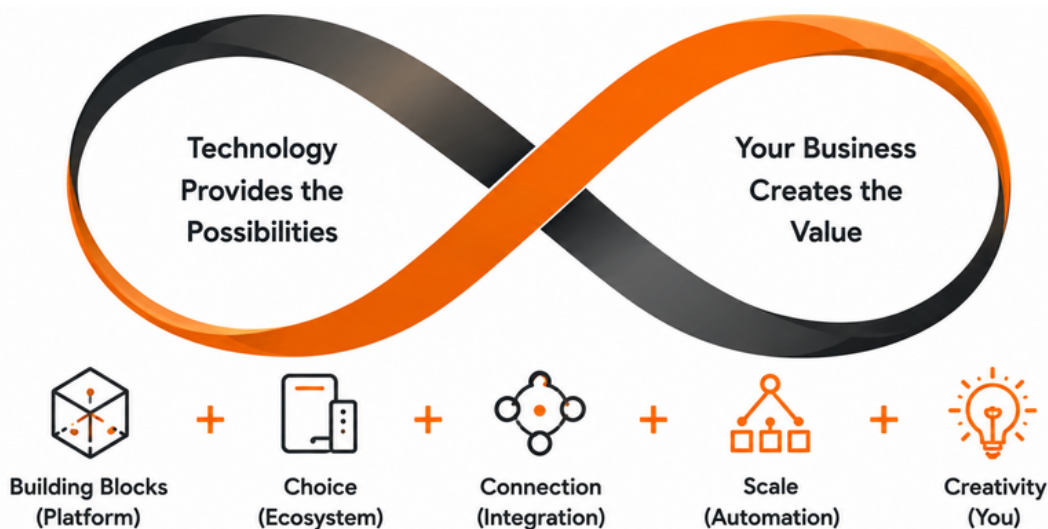


12. The Last Word:

Your Business Is the Product

The purpose of a modern OSS/BSS platform should not be to make every broadband provider operate the same way. It should make it easier for each provider to turn its unique knowledge, processes, relationships, network, data, and ideas into differentiated services.

The platform supplies the **building blocks**. The business supplies the **creativity**. The ecosystem supplies the **choice**. The integration layer supplies the **connection**. Automation supplies the **scale**.



Our businesses, networks, customers, and offerings are all unique.

That is precisely why we need platforms capable of making:
 $1 + 1 + 1 > 3$

Not by forcing everything into one product. But by allowing many different things to become one better business.

Indeed, a whole is greater than the sum of its parts.



13. About ETI Software Solutions

ETI Software Solutions helps broadband providers connect the systems, data, applications, and network technologies that power their businesses.

Integrate provides an open orchestration foundation that connects systems, data, applications, and network technologies across heterogeneous broadband environments.

Rather than requiring providers to replace the technologies that already work, Integrate helps those technologies work together—creating the shared data and operational foundation needed for automation, observability, and AI.

About the Author

David Tidd - Vice President, Engineering

David loves software development and software defined networking. His engineering team focuses on helping telecom companies automate and monETIze their network infrastructure. He lost his hair at the University of Missouri-Rolla now known as Missouri University of Science and Technology. He prefers to speak softly and laugh loudly, enjoying cooking, fine dining and traveling with his wife.



Have more questions? Contact us.

Email: jboozar@etisoftware.com or
priscilla.berarducci@etisoftware.com

Website: etisoftware.com

integrate[™]