

How MXL Connects More Than Media

Seven paradoxes shaping the next era of media production, Article 2

By Ronny van Geel, Director Product Marketing at Grass Valley



In this series, we explore the technologies, architectures and operational realities shaping modern media operations. Along the way, we'll examine how these individual pieces contribute to a larger operational picture and ultimately help organizations build a Unified Media Operation.

In our previous article, we explored why adaptability has become one of the most important characteristics of modern media operations. As audiences, business models and production requirements continue to evolve, organizations need the freedom to adapt without continuously rebuilding the foundations beneath them. But adaptability introduces a new challenge: as workflows become more distributed, software-defined and interconnected, how do independent systems continue to work together without limiting future choices?

The second paradox: Independent decisions require shared foundations.

For years, the industry has been told that interoperability matters. That's true, but interoperability alone has never solved a business problem.

The media industry has never had more choice. Organizations can deploy resources where they make the most sense, combine technologies from multiple suppliers and introduce new capabilities without rebuilding entire facilities. Dynamic Media Facilities make that possible by allowing operations to become far more adaptable than traditional infrastructure ever could.

Yet that adaptability introduces an interesting contradiction. Every media organization wants the freedom to make decisions that reflect its own priorities, workflows and business objectives. The ability to choose the technologies that best fit a particular operation is one of the most compelling promises of software-defined production. At the same time, the more freedom organizations gain to shape their operations, the more important it becomes that systems can work together in ways that preserve context, intent and meaning.

At first glance, those goals appear to pull in opposite directions. Dynamic Media Facilities encourage organizations to think independently, yet they depend on an increasingly shared understanding of how resources interact. The industry wants freedom of choice, while software-defined operations require a common foundation upon which that freedom can exist.

That tension sits at the heart of one of the most important challenges facing modern media production.



The value of open standards is not that they are open. The value is that they preserve future options.

The Real Challenge Isn't Connectivity

The media industry has spent decades becoming exceptionally good at moving signals. SDI remains one of the most efficient and deterministic ways to connect devices operating in close proximity, providing predictable performance with remarkably low overhead.

SMPTE ST 2110 expanded those capabilities into the IP domain, enabling organizations to transport video, audio and data across networks with far greater flexibility across facilities and locations. Both technologies continue to play essential roles throughout the industry and will continue to do so for many years to come.

The challenge emerging inside Dynamic Media Facilities and Software-Defined Production environments is fundamentally different.

As production capabilities become software *media functions*, media increasingly moves between processing tools that may reside on different compute nodes, clusters or platforms. In that environment, the question is not only how to transport a signal from one place to another, it is how software functions can access, process and share media in a way that reflects how compute environments actually work.

Traditional real-time media transport is largely linear and packet-paced. That model remains essential for many parts of live production, particularly at the boundaries between devices, facilities and operational domains. But inside a software-defined production environment, media functions often need a different model: one that reduces unnecessary copying, avoids repeated transport implementations and allows multiple functions to

access media through a shared fabric. This is where many software-defined environments encounter friction. If every application brings its own approach to media transport, every workflow can become a chain of conversions, translation layers and duplicated integration work. The result may be technically connected, but still inefficient, difficult to scale and harder to evolve.

A recurring theme throughout discussions with the GVx Council, echoed by industry leaders such as Brad Cheney, was the need for more optimization and less conversion. That principle sits at the heart of the MXL discussion.

The challenge is not simply moving media. The challenge is giving software media functions a shared way to access and exchange media efficiently, while preserving the timing, identity and context that make the workflow meaningful.

Enter MXL

The Media eXchange Layer, or MXL, was created to address this challenge at the level where software-defined production actually operates.

MXL moves media exchange inside a Dynamic Media Facility away from a linear, packet-paced transport model and toward a shared media access model. Instead of every application sending, receiving, copying and converting media through its own transport approach, MXL provides a common layer through which containerized media functions can access the media they need.

That matters because compute does not work best when every function behaves like a traditional device on a signal path. Compute works best when data can be accessed, processed and shared efficiently. MXL reflects that reality.

In practical terms, MXL creates part of the shared media fabric that allows DMF workflows to become more modular. Core production capabilities can be broken into containerized media functions that read from and write to a common exchange layer rather than each rebuilding its own media transport logic. Instead of multiple software applications each implementing their own approach to real-time transport inside the compute environment, they can share one exchange model designed for software-defined processing.

That does not make SDI or SMPTE ST 2110 less important. Each technology serves a different role. SDI remains highly efficient and deterministic for many device-level workflows. SMPTE ST 2110 remains essential for real-time IP media transport between systems, facilities and operational domains. MXL addresses a different challenge: how software media functions inside a DMF exchange media efficiently once the operation

moves into compute. This is why MXL is more than a conventional interoperability layer. It is a shared media access layer for software-defined production.

Metadata, timing, identity and APIs still matter deeply. They are what allow media functions to understand what they are accessing, how it relates to the workflow and how it should remain aligned with the wider production context. But they are not the whole definition of MXL. They are part of what makes the shared media fabric usable, meaningful and operationally coherent.

That is where **freedom** enters the story. When media functions share a common exchange model, organizations are less dependent on custom integrations, duplicated transport logic and workflow-specific workarounds. They gain more room to combine capabilities, introduce new functions and evolve workflows without rebuilding the foundations every time.

The Industry Perspective

The need for this kind of shared foundation is becoming increasingly clear throughout the industry. As organizations embrace Dynamic Media Facilities and software-defined operating models, a deeper question begins to emerge.

? *If workflows increasingly depend on shared foundations, who controls those foundations?*

At first glance, that may seem like a technical question. In reality, it quickly becomes a strategic one. The more software-defined media production becomes, the more the underlying frameworks, interfaces and standards begin to influence how organizations can evolve, integrate new capabilities and adapt to future requirements. That is why

the conversation around MXL quickly becomes a conversation about openness.

The value of open standards is not that they are open. The value is that they preserve future options. They allow organizations to continue evolving their operations, integrating new technologies and adapting to changing requirements without being forced into a single path.

As Emili Planas observed through the GVx Council, “MXL enables potential openness, but business models may limit actual portability.”

That distinction is critical. A standard can create the conditions for interoperability, but it cannot guarantee freedom of choice.

Ultimately, the value organizations realize from shared foundations depends on how vendors, partners and customers choose to build upon them.

For Grass Valley, that principle has always been central. The goal is not to create ecosystems that force customers into a particular way of working. The goal is to create environments that allow customers to define their own workflows, operating models and competitive advantages while retaining the flexibility to evolve over time.

Why Grass Valley Helped Shape MXL

Grass Valley's involvement in MXL extends far beyond adoption. As a founding contributor and Implementor-Chair of the MXL initiative, Grass Valley has helped shape both the direction of the standard and its practical implementation.

Working with organizations including the EBU, NABA and the Linux Foundation, Grass Valley has contributed to SDK development, interoperability initiatives and real-world deployments designed to validate how software-defined ecosystems can share media efficiently between Media Functions without introducing new forms of vendor dependency.

That involvement did not begin with MXL itself. Long before Dynamic

Media Facilities became an industry-wide initiative and before MXL was formalized, AMPP OS was already built around many of the architectural principles that MXL now standardizes.

At the heart of the AMPP OS media exchange architecture sits FrameCache, the shared media fabric through which applications read, write and share resources. Writers publish media into a shared environment. Readers consume only the resources they require. Timing remains synchronized through a common epoch-based reference.

Multiple FrameCaches can be synchronized to create larger processing environments while maintaining consistency across the operation.

Native MXL support interfaces directly with that fabric rather than through an external translation layer. Instead, it becomes a direct path into the exchange architecture that already powers AMPP OS. The result is a deeply integrated approach that allows media functions to share media through the same compute-native exchange principles that underpin AMPP OS, while preserving the timing, identity and APIs needed for software-defined workflows to remain meaningful.

This is not about supporting a standard because it exists. It is about helping advance an industry direction that aligns with how modern media organizations increasingly need to operate.

The Shift Is Already Underway

For Grass Valley, MXL is not a future ambition. It is already part of the operational reality of AMPP OS. Every AMPP OS deployment operating today already supports MXL natively. Because MXL interfaces directly with the media exchange architecture that underpins application interaction across the platform, it is not an add-on, a translation layer or a later adaptation. It is a natural extension of architectural principles that have been part of AMPP OS for years.

That reality matters because the industry is increasingly moving in the same direction. Organizations are building workflows around software-defined resources, distributed processing and applications that need to work together across organizational and technological boundaries. AI processing, remote

contribution, cloud-based services and hybrid production models are accelerating demand for richer interoperability and deeper integration between systems.

That demand is one of the reasons MXL continues to gain momentum across the industry. Organizations such as the EBU and CBC have played important roles in helping shape and advance the standard, ensuring that it reflects real operational requirements rather than theoretical architecture. Their involvement reflects a broader recognition that software-defined workflows require more than media transport alone.

The growing ecosystem around MXL demonstrates why this matters. Alliance Partners such as Zoom and TVU can leverage native MXL capabilities within AMPP OS to

create richer interactions between systems, enabling workflows where media functions can share media through a common fabric rather than through layers of custom transport and integration work.

For customers, the significance is straightforward. Investments made in AMPP OS today are already aligned with the direction in which the industry is moving, providing a clear path toward increasingly open, interoperable and software-defined operations.

Taken together, these developments demonstrate that MXL is no longer simply an emerging standard. It is increasingly becoming part of the infrastructure that enables modern software-defined operations to function in practice.



This is not about supporting a standard because it exists. It is about helping advance an industry direction that aligns with how modern media organizations increasingly need to operate.

In The End, It Is About The People

The real value of MXL is not that applications can exchange media more efficiently. It is that people can think less about how systems interact and more about what they are trying to accomplish.

Every translation layer, custom integration and loss of context introduces friction into an operation. That friction may be invisible when workflows are simple, but it

becomes increasingly significant as organizations adopt software-defined resources, distributed processing and applications from multiple vendors.

By providing a shared media fabric for software-defined production, MXL allows teams to operate at a higher level of abstraction. Instead of spending time managing the boundaries between transport

models, applications and custom integrations, they can focus on designing workflows, serving audiences and creating value.

The goal is not interoperability for its own sake. The goal is to make complexity disappear so that people can focus on outcomes rather than infrastructure.

What MXL Really Enables

So, what does MXL really enable?

The simple answer would be interoperability. But that would miss the larger shift. MXL enables software-defined production to work more like software-defined production.

It gives containerized media functions a shared way to access and exchange media, instead of forcing every application to recreate its own transport model. That reduces unnecessary conversion, avoids duplicated integration work and makes it more practical to build modular workflows from independently developed functions.

That is where freedom becomes practical.

When media functions can share a common exchange layer, organizations gain more freedom to combine capabilities, introduce new functions and evolve workflows without rebuilding the foundations every time. A production capability can be added because it creates value, not because an entire chain of custom transport and integration work has been solved around it.

The freedom MXL enables is therefore not abstract. It is practical.

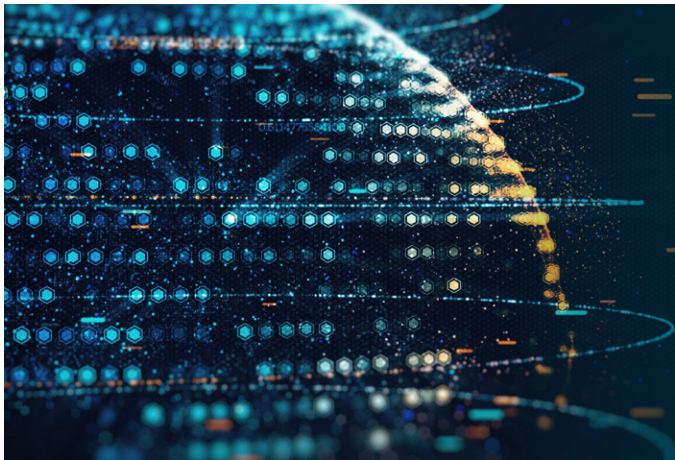
It is the freedom to choose technologies that best fit a particular operation. The freedom to build workflows from modular functions. The freedom to preserve investments while continuing to evolve. And the

freedom to keep future options open as production models, audience needs and business opportunities continue to change.

The most adaptable operation is not the one that predicts which technology will win. It is the one that remains free to adopt the right capability when it becomes valuable.

That is why MXL matters.

Not because it is open for its own sake. Because a shared media fabric helps preserve the freedom to keep evolving.



- *Freedom to choose technologies that best fit a particular operation.*
- *Freedom to build workflows from modular functions.*
- *Freedom to preserve investments while continuing to evolve.*
- *Freedom to keep future options open as production models, audience needs and business opportunities continue to change.*

That is why MXL matters.

Closing Thought

As we have seen, MXL is about more than moving media between applications. It provides a shared media access layer that allows software media functions to work together more efficiently, while preserving the timing, identity and context needed for meaningful workflows.

In many ways, that is what Dynamic Media Facilities have been searching for all along: not simply more

technology choices, but the freedom to make those choices without constantly rebuilding the foundations underneath them.

That is what MXL really enables.

As software-defined operations continue to grow, however, another question naturally follows.

If more of our workflows, data and operational decisions depend on shared digital foundations, how do

we ensure those foundations remain secure, resilient and worthy of the trust we place in them?

That is the challenge we will explore in the next article.

This product may be protected by one or more patents. For further information, please visit: www.grassvalley.com/patents

WP-PUB-3-1109A-EN

Grass Valley®, GV®, GV Grass Valley®, and the Grass Valley logo are trademarks or registered trademarks of Grass Valley USA, LLC, or its affiliated companies in the United States and other jurisdictions. Grass Valley products listed above are trademarks or registered trademarks of Grass Valley USA, LLC or its affiliated companies, and other parties may also have trademark rights in other terms used herein. Copyright © 2026 Grass Valley Canada. All rights reserved. Specifications subject to change without notice.

www.grassvalley.com Join the Conversation at GrassValleyLive on [Facebook](#), [X](#), [YouTube](#) and Grass Valley on [LinkedIn](#)