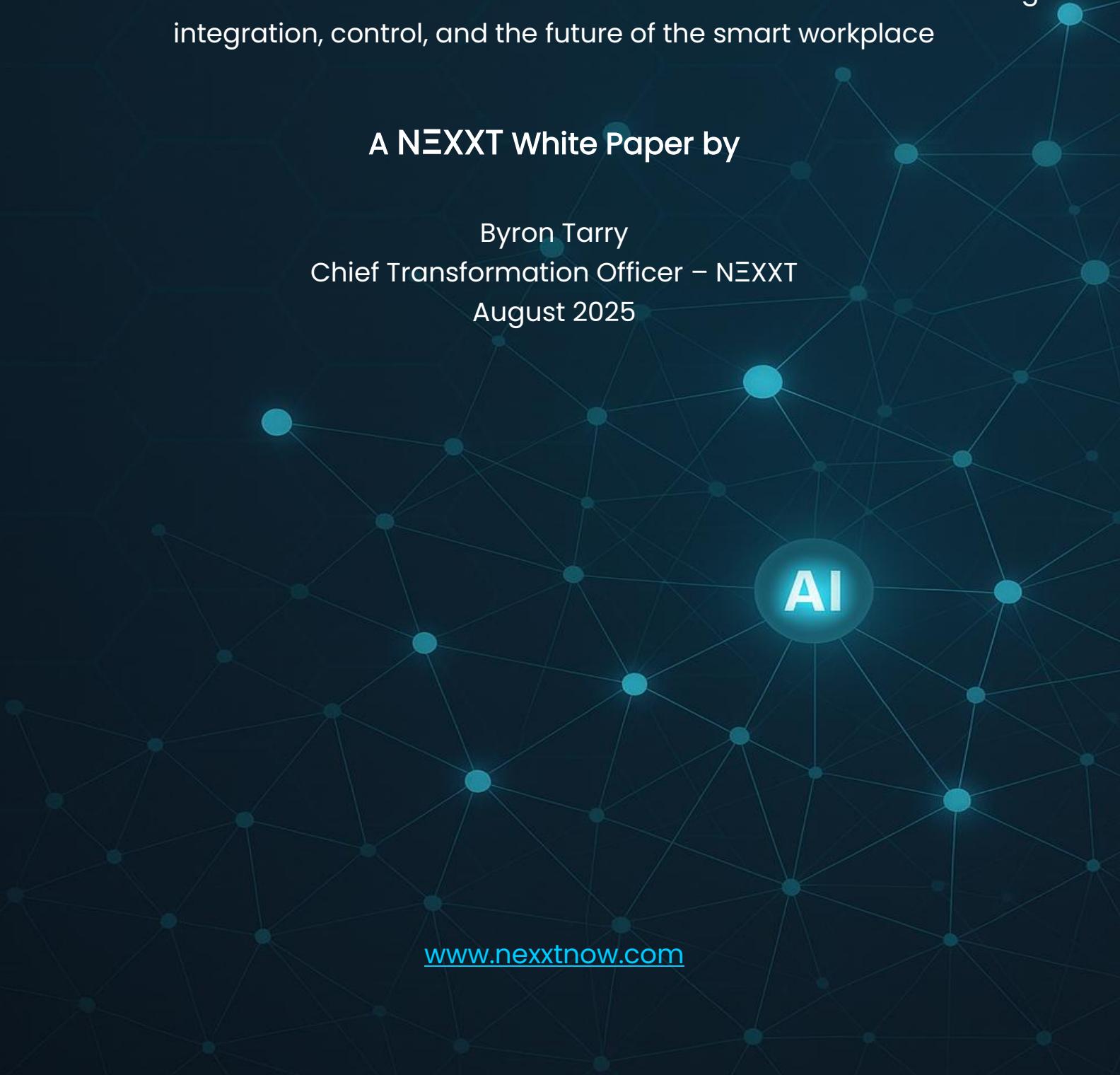


The Rise of MCP in AV

From Command to Context: How intent-driven orchestration is redefining integration, control, and the future of the smart workplace

A NEXXT White Paper by

Byron Tarry
Chief Transformation Officer – NEXXT
August 2025



AI

www.nexxtnow.com

The Rise of MCP in AV

From Commands to Context: *How intent-driven orchestration is redefining integration, control, and the future of the smart workplace.*

Introduction

There are moments in technology when the ground seems to move all at once. Sometimes, these moments are quiet – like the shift from serial cables to Ethernet; sometimes, they are thunderous, like the rise of the cloud or the arrival of the smartphone. Today, as artificial intelligence and intent-driven systems transform how we interact with information and each other, the AV industry stands at the edge of another foundational shift.



For decades, the way we have controlled AV systems has felt logical – almost ritualistic. Control meant writing scripts, wiring touch panels, mapping “if this, then that” logic against every possible scenario. In these worlds, every device was a kingdom, every command a dialect, and programmers – often the unsung heroes – were the translators, building brittle bridges across a patchwork of APIs and proprietary codes.

But cracks are showing. As every system and space becomes smarter, more dynamic, and more open, the cost of maintaining those bridges has grown – sometimes quietly, sometimes painfully. Firmware updates that threaten to break what was working yesterday. Features left on the table, simply because upgrading one piece risks the whole. Technical debt mounting with every workaround. And amid all this, end-users and organizations expect more – more personalization, more reliability, more agility, more value.

That's why the emergence of the **Model Context Protocol (MCP)** is not just another integration milestone. It's a philosophical reset. MCP does not merely promise better APIs. It replaces the premise of “command and control” with one of “orchestration and intent.” In an MCP world, what matters is not the string you



send, but the outcome you want. The bridge is built, in real time, by systems that understand each other's capabilities, context, and constraints.

Over the course of recent industry roundtables and workshops, leaders from across AV, support, and enterprise IT agreed: this isn't a change at the margins. It is a change in the center of gravity. As support platforms like Xyte, Utelogy, and Netspeak gain the ability to sense, adapt, and increasingly control, the traditional boundary between "support" and "user experience" is dissolving. The next control system may not look or feel like anything we've installed before.

This white paper/perspective is a product of that dialogue and this moment in time. It draws on decades of lessons – hard-won by programmers, integrators, designers, and end-users – and looks ahead to a world where integration is not just easier, but fundamentally different. It is a call for readiness, creativity, and leadership at every level, from executives to system designers.

While this paper will focus on AV and UC as the relevant and immediate proving ground, MCP's implications extend well beyond – into the broader smart workplace and even adjacent industries. What follows will take you on that journey: beginning with a clear definition and technical grounding of MCP itself, moving into its practical impact on AV and UC control, extending outward into the opportunities of the smart workplace, and finally reflecting on the organizational implications – from experimentation to role transformation – that every stakeholder will need to consider.

Understanding MCP: Beyond Technical Specifications

So, let's set this up with some definition of Model Context Protocol (MCP) itself. MCP is not just another interoperability tool. It is a framework for adaptive orchestration – designed for environments where context changes constantly, requests are unpredictable, and systems must respond with agility rather than rigid code. MCP grew out of the need for AI systems to connect with the right context, but its implications stretch well beyond that. It enables any platform, device, or agent to not only identify what functions exist, but also to understand under what conditions they matter, and how to apply them in the moment.

Traditional APIs and control systems operate like static directories: useful if you know the exact "call number" and if the world around you doesn't shift much. But MCP is built for a more dynamic reality. It provides the scaffolding of rules, policies, and structure – while also giving agents the autonomy to interpret, adapt, and act in real time.

Certainly I encourage you to use AI to teach you more about MCP, but here are a couple of human analogies that might help as some basic concepts:



Analogy 1: The Air Traffic Controller

Picture a control tower at a busy airport. Planes are constantly arriving and departing, weather patterns are shifting, and runways open or close unexpectedly. Controllers don't simply follow a script — they interpret conditions, weigh priorities, and coordinate in real time. They draw from established procedures, but also from judgment, experience, and context. Their role is not to fly the planes, but to ensure the whole system stays safe, efficient, and responsive to change. MCP plays a similar role across digital environments: discovering what's available, sequencing actions, and adapting dynamically as conditions evolve.

Analogy 2: The Executive Assistant

Now think of a world-class executive assistant managing the inflow of requests for multiple leaders. Their job is not to forward every call or schedule every meeting by rote. Instead, they filter, clarify, and prioritize. They know when to pass something through immediately, when to decline, when to ask for more detail, and when to act on behalf of the executive without even disturbing them. They take guidance from high-level priorities — “this quarter we’re focused here,” or “these relationships matter most” — and apply those principles dynamically to unstructured, real-time demands. MCP functions in much the same way: it filters noise, interprets intent, and orchestrates resources with autonomy, ensuring that systems act intelligently rather than mechanically.

Technical Explanation

With these analogies in mind, we can move into the technical explanation of how MCP actually operates. So what happens under the hood when MCP is in play? Let's walk through it in the flow of a real request with a bit of an AV/UC lens.

1. The Host: Making the Request

This could be an AI, an agent, or even a desktop app. Think of it as the executive making a request: *“Prepare this room for a Teams meeting.”* The host doesn't spell out every step — it just expresses the outcome it wants.

2. The MCP Client: Opening the Channel

The client acts like the secure line that connects the request to the environment. It makes sure the conversation happens in the right place, with the right protections, so nothing gets lost or misheard.

3. The MCP Server: Surfacing Capabilities

Here's where the magic happens. Sitting in front of a device, a database, or a file system, the server is like the assistant or controller who knows exactly what resources are available right now. It doesn't just list “commands.” It surfaces richer context: what the device can do, what state it's in, what permissions apply, and even sequencing advice.

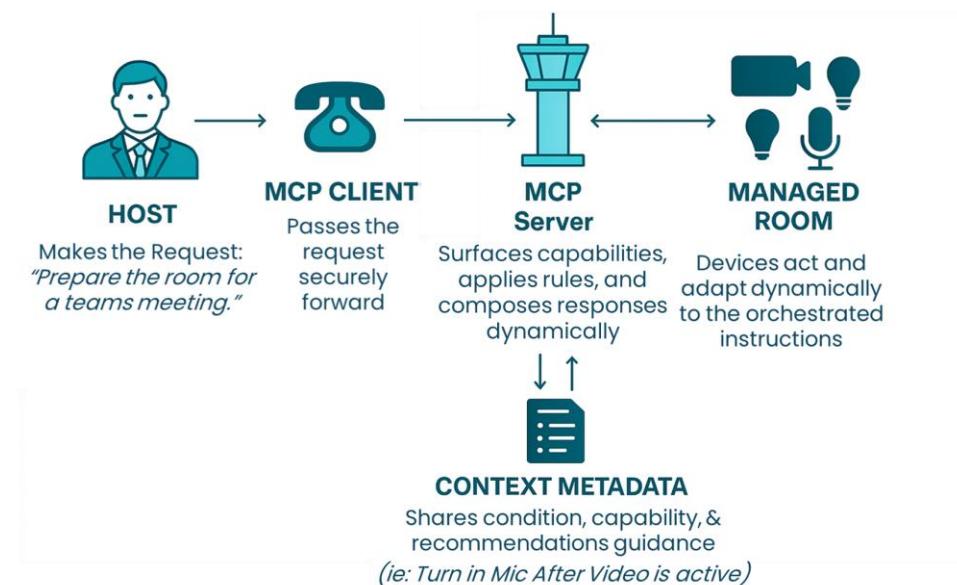


4. Context Metadata: Adding Intelligence

This is what transforms MCP from a simple switchboard into an orchestrator. Devices can share not just their functions, but also their conditions and best practices — like a microphone system that says, *“Turn me on after the video feed is active for best performance.”* With this kind of metadata in play, MCP-enabled agents don’t just act, they act intelligently, sequencing things the right way and adapting to what’s happening in real time.

This structure dramatically reduces brittle code and one-off integrations. Instead of coding every possible workflow, you publish or deploy an MCP server once — and every compliant agent can leverage it.

While the simplified diagram following shows a single MCP client and server for clarity....



...real deployments usually involve **many clients and many servers**:

- **Multiple Clients:** Any agent that wants to *use* device capabilities acts as a client. This might include a user interface in the room, a support/monitoring platform, or even an automation script.
- **Multiple Servers:** Each device or subsystem that wants to *expose* its capabilities is an MCP server (camera, microphone, DSP, lighting, etc.).
- **Hybrids:** Some systems can play both roles. For example, a room “gateway” may expose a unified server interface upstream, while acting as a client downstream to individual devices.

The **key value of MCP** is that you don't have to hand-code how these pieces talk to each other. The protocol negotiates context dynamically, so whether you have one client and one server – or many – the orchestration still works.

Key Shifts Compared to Traditional APIs

Aspect	MCP	Traditional API
Mindset	"Bring me whatever context I need to fulfill intent."	"Call this exact endpoint for a result."
Statefulness	Context-aware; tracks conversations or tasks	Stateless; every call is independent
Flexibility	One adapter can expose multiple tools or services	Separate API for every system
Maintenance	Adaptive; fewer breakages on updates	Fragile; one string change can break flow

Why this matters: In an MCP world, you no longer have to rewrite logic every time a device updates. If a device's capabilities expand, the agent discovers and uses them. If a device is missing, the agent finds an alternative. The system becomes dynamic, resilient, and context-driven – much more like a skilled operator than a rigid program.

API to MCP: Shifting the Integration Paradigm

In virtually every digital ecosystem today, APIs serve as the common language that lets systems communicate – whether that's a single app connecting to a database, or entire networks of devices and platforms exchanging information. They have become the foundational mechanism for integration across industries, from finance and healthcare to IT and AV – allowing control systems, applications, and devices to speak to each other through pre-defined endpoints and payloads. APIs offered a way out of the hardware lock-in and “one protocol per device” age. They made automation and remote management possible. But they also brought new limitations: each API was a new language to learn, and a new source of risk every time something changed.

The “Key for Every Door” Problem



In the traditional model, APIs are like having a ring with dozens of keys – one for each device, each room, each function. As your system grows, so does the keyring – and so does the headache of maintaining, updating, and troubleshooting every lock.

- Every firmware upgrade brings anxiety: “Will the key still fit?”
- Adding features often means re-writing code or negotiating new APIs.
- Even minor updates (a field change, a renamed function) can break everything downstream.
- As a result, upgrades are often delayed, systems remain static, and the risk of technical debt rises year after year.

MCP: The “Master Key” for Dynamic Environments

MCP replaces this brittle architecture with a “master key” – one that adapts to the room, understands the context, and negotiates its way to the right outcome, rather than rigidly unlocking a single door.

- An MCP agent asks not just “Can I open this lock?” but “What doors are available, who’s in the building, and what’s our purpose today?”
- If a device adds new features, the MCP agent can discover and use them immediately—without recoding.
- If a device is upgraded or swapped out, the agent can adapt, using metadata and context to select alternatives or offer new experiences.
- The system becomes both more resilient (less breakage), and more agile (faster to evolve).

Reliability, Upgrades, and Security

This is more than technical elegance – it’s a fundamental shift in how organizations can manage risk. In API-centric systems, the fear of “breaking the flow” causes teams to postpone upgrades – leading to outdated firmware, unpatched security holes, and slow adoption of new features.

With MCP:

- Systems can be upgraded continuously and confidently, knowing context-aware negotiation will fill gaps and prevent routine breakage.
- Security updates are less likely to cause catastrophic integration failures.
- New devices and capabilities can be piloted or rolled back with minimal disruption.

The Mindset Shift

The move from API to MCP is not just a technical leap – it’s a change in mentality:

- From “call and response” to “goal and negotiation.”



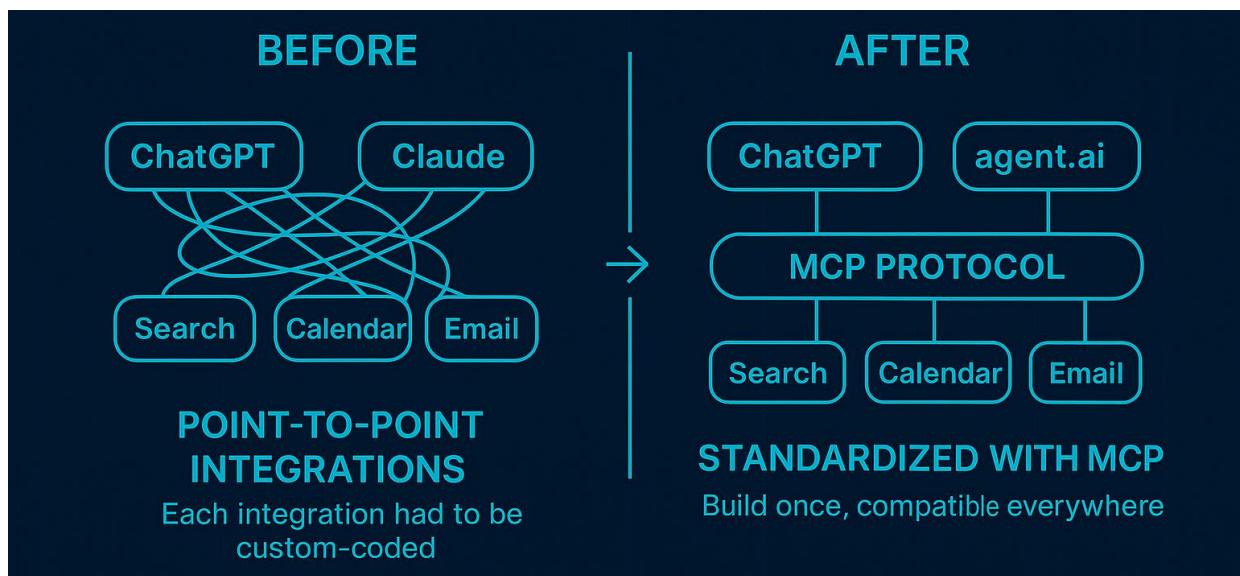
- From “stateless function” to “stateful orchestration.”
- From “hoping it doesn’t break” to “trusting it to adapt.”

In an MCP-enabled world, the role of the integrator, the programmer, and even the end user changes. The conversation is less about “which string do I send?” and more about “what are we trying to achieve, and how can the system get us there?”

The Future of Control – Support Platforms as the New Control Layer?

Like AI more broadly, MCP is not a narrowly bounded innovation – its potential applications stretch across industries and environments wherever systems need to adapt, negotiate, and orchestrate dynamically.

At its core, MCP was built to manage the ubiquity required for intercommunication between what will be a tsunami of “agentic” AI tools, applications, platforms, etc. Especially in an AI age where even human language barriers are being broken every day, any human is foundationally able to communicate with another and have an interactive conversation with another, providing context, expectation, capability, etc. So think of MCP as a starting point being able to tackle what have been struggles to get different LLM’s talking to different workflow tools. Now think of that same language translator for humans applying to interface between LM and application, or even between LLM’s. Just pause for a moment and consider what some of your current AI frustrations are, despite the variety of tools making this ever easier still how daunting it is to try and get your day to day LLM to interface with the rest of your day to day applications so it has depth and breadth of context beyond your prompts, and how that might be transformationally changes moving forward with MCP.



But to make this discussion concrete, it helps to focus on where the impact is likely to be felt first and most acutely: in the AV and Unified Communications realm.

So, what will all of this ultimately mean for who owns, defines, and delivers the control experience in the AV real me all know so well. MCP is not about just another technical upgrade – it's about embracing a mindset shift, a strategic enabler, and an organizational cultural imperative that reframes what's possible for integration, control, and user experience. Every stage of MCP's evolution nudges us away from the brittle, command-centric world of APIs and hardware-defined boundaries – and toward a more adaptive, outcome-driven paradigm.

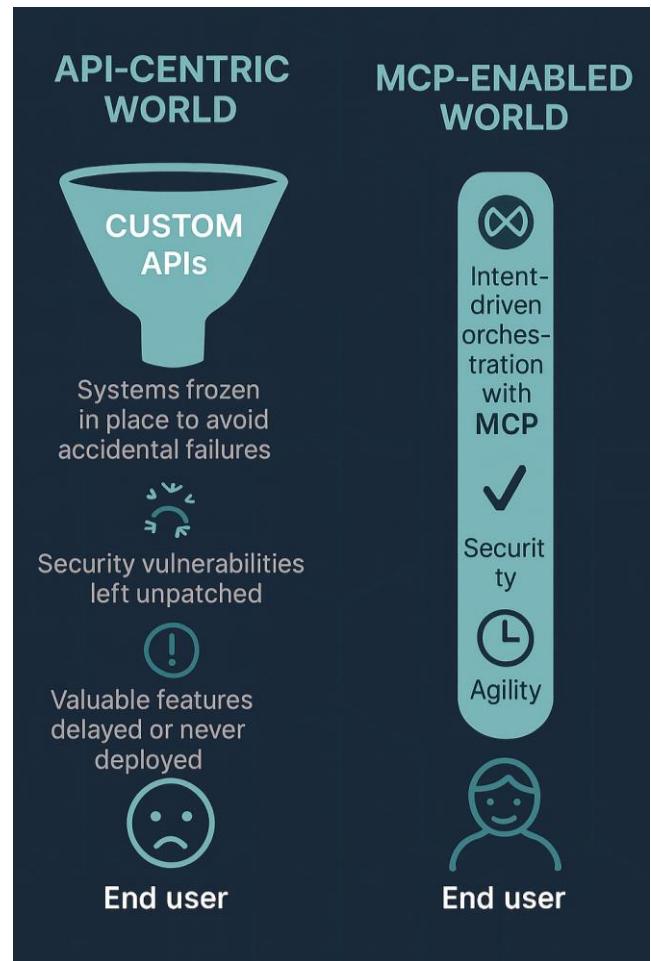
Here's one important potential layer of this shift:

Firmware and API fragility has become one of the silent saboteurs of modern AV agility. One of the most persistent sources of risk in AV systems is the concern that a firmware upgrade might subtly change an API string or behavior, potentially breaking a critical control sequence. This has made integrators cautious – often postponing updates, running outdated versions, and manually regression-testing environments that should otherwise benefit from modern enhancements.

This friction leads to:

- Systems frozen in place to avoid accidental failures
- Security vulnerabilities left unpatched
- Valuable features delayed or never deployed

Avoiding upgrades doesn't just create friction – it accumulates risk. Unpatched firmware exposes enterprises to cybersecurity vulnerabilities, compliance failures, and integration breakdowns. MCP doesn't just support agility – it safeguards reliability.



MCP, through its context-driven, intent-oriented architecture, effectively neutralizes this pain point. With a shared understanding of capabilities and context between systems, the communication no longer depends on perfectly matching syntax. Upgrades can be embraced with confidence, knowing that if a device adds new functionality or slightly modifies behavior, the MCP interface can absorb and adapt to those changes without disrupting service.



So, at first glance, MCP's strongest utility appears to be on the support front – improving fault detection, automating resolution, and streamlining multi-vendor management. And that's where platforms like Xyte are gaining traction: by building foundational control visibility for support and operations teams.

But here's the critical observation: once a support platform can reliably monitor, access, and autonomously correct the behavior of devices... what's left for the "real" control system to do?

The logical conclusion of this journey leads us to a provocative, even uncomfortable, observation: maybe the future of control may not belong to the traditional control system players at all. This may sound bold but consider this proposition: With MCP the trigger to what sets this in motion, perhaps the next generation of control systems won't come from Crestron, AMX, or Q-SYS! Perhaps they'll come from platforms like Netspeak, Xyte, and Utelogy – platforms currently presenting as 'support' tools.

While that doesn't mean that the traditional players won't adapt, let's just explore the above proposition for its "transformative" premise. If you already know what a room should be doing—and have the authority to make it happen – you don't just own the support experience. You're one interface away from owning the user experience too. The boundary between "support" and "control" erodes with every software update and every new integration, and it's only a matter of time before it's orchestrating everything else – especially when user intent is expressed as "outcomes," not button presses.

This transition isn't just speculative – it's happening now, with the likes of Netspeak and Xyte already issuing live commands and orchestrating outcomes in real environments, often with a traditional control processor still in place. The difference? The locus of intelligence and value is moving upstream: from the hard-coded command, to the adaptable agent, to the ecosystem that learns and evolves.

The "opinion" here is not just to spark a generalized debate – it's to challenge every leader, integrator, programmer, and designer to recognize that the skills, investments, and cultural models that created yesterday's control systems may not be the ones that ensure leadership tomorrow. The traditional model is crumbling. If MCP makes constant, agile improvement possible, then the value of your system becomes less about what it does at install—and more about how it evolves.

In short: control by correction might just be evolving into control by default.

Certainly, this isn't the end of the control system. But it is the beginning of the end of control as we've always known it. The question is: will you lead, or will you be disrupted?

From the Room to the Workplace: MCP and the Smart Environment

Having examined MCP's disruptive potential in the control layer, it's worth asking how these same dynamics extend when the room itself becomes part of a much larger workplace ecosystem. If MCP challenges our assumptions about control at the room level, its implications for the broader workplace



are even more profound. We all know that until now, most AV and UC control has lived in a fairly narrow context: the room, perhaps in more recent time a network of rooms within a building or estate. Integrations with building systems – lighting, HVAC, access control – have existed, but often in limited or simplified forms. Rarely could we claim that AV systems were a true part of the broader smart building ecosystem.

It should be becoming clear though that MCP changes that potential equation. By enabling systems to communicate through context rather than brittle, predefined scripts, MCP lowers the barrier to integrating across silos. Suddenly, interfacing with building systems does not require deep protocol expertise or tightly engineered workflows. Instead, MCP can negotiate outcomes dynamically: syncing environmental controls to meetings, optimizing energy use around occupancy, or coordinating building-wide responses to events.

In an era where the very “Why” foundations of the physical workplace (office) has shifted in a post COVID reality, this shift reframes a critical strategic question: *who owns the integrated “smart workplace” experience?* With MCP, the line between AV, IT, and building systems begins to blur. Whether through a single “master integrator” or through deeper collaboration between traditional silos, MCP opens the door to comprehensive, intelligent workplace orchestration.

The implications are profound. If the office is to remain a destination for collaboration and experiences that exceed the home office, MCP may be the connective tissue that allows every system – from conferencing to climate – to align in service of that goal. Knowledge is power, and the more context we enable MCP to carry across platforms, the more opportunities we unlock for alignment, creativity, and entirely new kinds of workplace experience design.

MCP Is Here Now: Early adoption and the bridge phase

Although the MCP specification is new, it is certainly not theoretical – it is already showing up at an ever more rapid pace in production roadmaps, developer guides, and open servers. While certainly it’s not prevalent in every system or platform you will find right now, you do not have to (and shouldn’t) wait for every endpoint to be “MCP native” before you start getting value from intent-driven orchestration.

As you might expect, one tangible pattern to cope with the fragmented MCP availability to this point is the bridge model. Platforms expose an MCP server upstream so agents can discover capabilities and act by intent, while configuring, managing, and speaking each vendor’s native API downstream until those devices ship with an MCP capability themselves. Xyte announced and demonstrated exactly this at InfoComm this year – an AV MCP Server layered on top of its Universal Device APIs – giving integrators an MCP entry point today while preserving reach across mixed fleets. And as those devices indeed adopt MCP, the API is simply dropped in favor of a new MCP interface.



Beyond the room, smart-building signals are already visible. Siemens Building X published a developer guide showing how to run an MCP server alongside its openness APIs so agentic workflows can navigate building data and actions. In parallel, community servers are wrapping established building and SCADA stacks behind MCP – for example BACnet-to-MCP and a WinCC Unified MCP server that fronts Siemens' GraphQL API for SCADA. These are early-stage, but they show a tangible and credible migration path: wrap what exists with MCP now, deepen native adoption over time.

Practically, this means two horizons. Short term – adopt an MCP bridge to unify orchestration, centralize policy, and insulate workflows from brittle API changes while vendors catch up. Medium term – expect more devices and subsystems to embed MCP servers directly, shrinking or removing the interpreter layer. Either way, the direction of travel is clear: MCP is operating here and now, not as a distant promise, and it offers a path to start learning, proving ROI, and scaling ahead of the curve.

MCP: Not Without Challenges

MCP's potential for agility and personalization is compelling, but to be clear, is not without trade-offs in the transition to the full potential of this new control model. The advantage of today's tightly scripted, universal control systems lies in their predictability: every input, every outcome, every workflow can be anticipated – at least in theory.

In the MCP era, as with much of AI, we invite systems to interpret, adapt, and increasingly to make decisions on our behalf. This brings the promise of frictionless experiences and richer personalization, but also introduces new layers of unpredictability. This potential for transformation certainly isn't just about handing over unlimited control; it's about shifting from rigid automation to collaborative orchestration – where AI becomes assistant, advisor, friction remover, and experience guide. And how best to achieve that long term potential will take time to understand how best to leverage.

As such, as it should, the journey from support-centric management to truly user-facing, adaptive control will be gradual and will demand experimentation, transparency, and user readiness.

Leveraging & Positioning for the future

The point of this entire discussion is in part around who are the parties best positioned to lead in this new landscape – whether that's in terms of specific roles, or organizations themselves? And I am essentially positioning that that service and support organizations, and their software platforms, have been architected for years around long-term evolution & optimization, responsiveness & proactivity, agility & change management, and relationship-based lifecycle value. And because of that, many are already weaving AI into their DNA.



By contrast, many traditional control-centric integration platforms & those who configure and deliver them have been focused on achieving static, day-one capability – often at the expense of flexibility or continuous evolution. This is the challenge and opportunity: platforms built for adaptability (think ServiceNow's evolution from ticketing to full-service ecosystem) are primed for MCP's world, while those still anchored to legacy architectures must choose to reinvent, or risk being left behind.

In short, as the foundations of control shift, the organizations that are built for continuous evolution – those who see themselves as partners in the user journey, not just installers of hardware – are poised to lead. The challenge for traditional control platform mindsets is not only technical, but cultural and architectural.



Relevance and Implications for Industry Stakeholders

These shifts are not about prescribing the future, but about preparing for and innovating/evolving for it: understanding both the opportunity and the learning curve ahead as AI and MCP reshape the foundations of control, and as new leaders emerge in the space.

The shift from API-driven logic to MCP-enabled, intent-driven orchestration is not just a technical transformation – it's a sweeping redefinition of roles, responsibilities, and value creation across the AV and integration landscape. We have talked about service implications, but it should be clear by now that MCP has the potential to reach deep into every part of the ecosystem: from the C-suite to the front lines.

If that is the case, to bring some practical relevance and representation to all readers, here are some important implications and considerations for even just a selection of other stakeholder personas:



Execs

Executives and Business Leaders: For executives – the strategy leaders of organizations at any point in the supply chain, MCP should become a strategic imperative. MCP allows organizations to be more agile – adopting new features, scaling across platforms, and creating differentiated user experiences without the friction and technical debt of brittle integrations. MCP is not just about operational savings; it's about unlocking new business models, enabling service-based offerings, and futureproofing against disruption. Executives will increasingly need to frame their direction and guidance towards answering questions like: *Are we building for flexibility? Are we ready to let intent, not infrastructure, define our technology roadmap?*





Product Managers

Product Managers: Product managers will need to think beyond feature lists and integrations to holistic, outcome-based experiences. With MCP, the focus shifts from shipping static feature sets to delivering adaptable, evolving solutions that get smarter as new devices, tools, and user needs emerge. Roadmaps become less about point-to-point integrations and more about exposing the right capabilities and metadata for AI agents and orchestration platforms to leverage. Competitive differentiation will rest on *how discoverable and composable your product is in an MCP ecosystem.*



Project Managers

Project Managers: For Project Managers, the transformation driven by MCP presents a unique challenge to traditional definitions of scope, deliverables, and completion. Historically, Project Managers in AV and IT inherited highly specified, tightly scoped work packages measured by well-defined criteria and a clear "done" state.

As technology shifts from hardware-centric to software-driven, and MCP turns static feature sets to agile, continuously evolving platforms, the classic boundaries of "project" begin to blur. In an MCP-enabled world, is a PM managing a one-time delivery – or are they overseeing the deployment of a living, breathing foundation for ongoing innovation? While the initial deployment may still involve tangible infrastructure and handover, if the true value comes in how the system can adapt and improve long after the "go-live" that will need a more "Program" than "Project" management-oriented approach and skillset.



Programmers

Programmers: The biggest shift may be here. Programmers will move from being coders of rigid workflows, to architects of intent-driven experiences. They'll need to spend less time cross referencing API strings and more time modeling user goals, context, and adaptive responses. Skillsets in metadata design, prompt engineering, and "context mapping" will become as valuable as traditional programming. A *composer of possibilities*, not just a builder of bridges. Add this to the other shifts in programming like AI/Vibe coding, voice replacing traditional visual GUI, and the programmer's role we know today will look very different very quickly.

As a side note, I'd suggest that there is tremendous opportunity for traditional "programmers" to become more engaged in the internal AI workflow implementation efforts – rather than integrating AV hardware, integrating business systems and data for AI & related tools. So, the evolution of optimal "programmer" roles and skillsets should be a critical path of opportunity and evaluation in the short to medium term.





System Designers

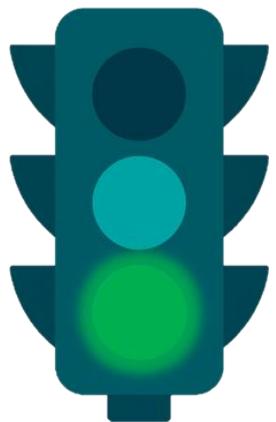
System Designers and Solution Architects: System designers are the translators between organizational goals and technical architecture. In the MCP era, their job will be more than just hardware/feature specification, but to ensure every device, platform, and service can (and will continue to) clearly expose its capabilities, metadata, and context, making the whole system *legible and orchestratable* by intelligent agents. Traditionally these resources have been closer to usage intent and outcome design than Programmers, but as AI programming becomes more prevalent, one could argue that an "intent definition" skillset will be the key "programming" skillset making these design roles more relevant to configuration and commissioning than any.

Acting Now – Leadership from Every Level

The opportunity around MCP is here, the time to act is now. And to be clear, to all readers, as with many other elements of AI driven transformation, leadership around MCP cannot and should not only come from the top down. Its impact cuts across executives setting vision, managers aligning resources, and frontline technologists experimenting in the details of day-to-day work. Proactivity is the key: do not wait to be told. Whether your mandate is formal or informal, this is a moment to step forward. Leadership in AI and MCP adoption must come from every facet of the organization.

That means executives framing strategy and culture around adaptability. It means project managers rethinking delivery as a living program, not a one-time handover. It means programmers, designers, and support teams being bold enough to test, to pilot, and to share results – even if the mandate has not yet been given from above. The future will be shaped just as much by the bottom-up energy of those closest to the systems as by the top-down direction of those shaping business strategy.

In the body of this paper we have focused on context, impact, and urgency. But for those who want to go further – to explore specific recommendations, step-by-step opportunities, and persona-based implications – we have included a set of appendices that expand on these themes. Practical guides for both strategists and practitioners who are ready to move from understanding to action.



Conclusion: A New Era for Integration, Control, and Leadership

The journey from APIs to MCP is not simply about protocol – it is about possibility. MCP is a macro-level shift with implications that extend across industries, environments, and even business models. But throughout this paper, we have zoomed in and out for a reason. To understand MCP in its full breadth, it helps to ground it in the place where its impact will first be felt most acutely: the AV and UC industry. And within that, in the very heart of our work – the control paradigm.

Why control? Because this is the daily arena where brittle APIs, firmware fragility, and static integrations create friction, cost, and risk. It is also the arena where MCP's adaptive, intent-driven orchestration can demonstrate immediate value – reducing breakage, enabling upgrades, and safeguarding user experience. That immediacy makes control not just an application of MCP, but a springboard: a financially viable, ROI-positive proving ground where our industry can experiment, learn, and lead.

And that matters far beyond AV. Once the concepts of context, negotiation, and orchestration become embedded in our practice, the lessons can extend outward – to broader workplace integration, smart building ecosystems, and ultimately to any domain where systems must collaborate dynamically. MCP is not “about AV,” nor “about control” – but these are the places where we can grasp it, test it, and translate it into broader impact.

There is, however, a flip side. Just as MCP allows AV and UC professionals to step outward into wider domains, it also lowers the barrier for adjacent industries to step inward. A building automation company may extend naturally into AV-rich experiences; a workplace IT platform may evolve into a meeting platform. This convergence is not hypothetical – it is already unfolding. Which is why urgency, experimentation, and innovation cannot wait.

MCP stands as both a technological and philosophical pivot. It empowers every stakeholder – from executives to programmers to end-users – to rethink their role in shaping technology, not merely reacting to it. But leadership here is a choice. Those who treat MCP as a niche standard will limit themselves to incremental gains. Those who treat it as a launchpad – a way to reimagine control, to expand into new arenas, and to invite entirely new kinds of collaboration – will be the ones to define and benefit from the next era.

The path forward is clear: learn, experiment, educate, and challenge assumptions. Start with what we know best – control. Use it as the springboard to build maturity, confidence, and organizational capability. Then look outward: to the smart workplace, to broader digital ecosystems, to entirely new business models. MCP is not only the next step for integration. It is the foundation for transformation – and it is ours to shape.



About the Author

Byron Tarry is the Founder and Chief Transformation Officer of NEXXT. A progressive leader with over 30 years of experience in the Audio Visual and Collaboration Technologies industry, Prior to NEXXT, Byron served for nearly a decade as CEO of GPA, a “modern collaborative economy” inspired global AV integration company. He brings a deep understanding of how technology & “think different” culture can transform businesses – and has a passionate belief in the potential to do so meaningfully, collaboratively, and sustainably.

That passion, combined with a “think different” mindset, laid the foundation for NEXXT’s unique positioning: an education, advisory, and enablement organization driven by a social enterprise mission, and inspired by the collaborative economy. Through NEXXT, Byron leverages his expertise in business model innovation to help organizations navigate the transformative forces reshaping today’s business environment – from global scale, to sustainability, and the disruptive power and potential of AI – to not just to survive, but thrive in this unprecedented era of change & opportunity.

About NEXXT

NEXXT is a purpose-built, community-powered initiative committed to enabling applied transformation in the AI era – particularly across the AV, collaboration, and modern workplace industries. With the accelerating pace of technological change, NEXXT serves as a trusted ecosystem to help individuals and organizations move from AI confusion to confidence, from passive reaction to active reinvention. It empowers those at all stages of their journey – from the AI-curious to the AI-courageous – to navigate uncertainty with clarity, connect with like-minded peers, and create tangible business impact.

Built on four core pillars – Structured Learning & Education, Consulting & Advisory, Digital Community, and Storytelling & Brand Elevation – NEXXT seeks to turn AI theory into real-world transformation. We blend practical education, curated community engagement, tailored consulting support, and narrative amplification to help individuals, teams, and industries take their next bold steps. More than just a platform or an organization, we are a movement: a collaborative place & space future-forward leaders can trust to learn, build, and grow – together.

To learn more about NEXXT, visit us at www.nexxtnow.com. Whether specific to this white paper, or on broader business transformation topics, if you’d like to discuss how NEXXT might be able to help you or your organization in an education, consultation, or brand elevation capacity, reach out via info@nexxtnow.com.



Appendix 1: What Should I/We Do to Embrace the MCP Opportunity

If these implications are indeed even partially valid, then successfully navigating the transition to an MCP-enabled future requires not just a technical understanding and vision, but deliberate and incremental action. So first let's take a look at some practical opportunities and imperatives to consider implementing to ready for this shift:

1. Create MCP Champions

Identify and empower individuals within your organization who demonstrate both technical fluency and a capacity for strategic thinking. These MCP Champions will lead internal education, experimentation, and pilot projects. They should be given explicit mandate and executive support to collaborate across teams, share learnings, and serve as the first port of call for new MCP-related opportunities or challenges.

2. Build Education and Awareness

MCP is new, but it won't stay niche for long. Build broad-based understanding through workshops, lunch-and-learns, external speakers, or "MCP 101" resources for everyone from executives to programmers. Promote cross-functional discussions – product managers, designers, integrators, support staff, and business development should all know what MCP enables and why it matters.

3. Strategic Assessment

Assess your current systems, workflows, and value propositions through the lens of MCP. Where are you most vulnerable to brittle integration? What manual or static processes could be automated, made adaptive, or opened up to AI orchestration? Which devices and platforms would benefit most from MCP compatibility—either as hosts, servers, or both?

4. Launch Pilot Projects and Experimentation

Don't wait for a perfect use-case. Start with focused pilots – perhaps a meeting room, a support workflow, or a managed service integration – where MCP can be introduced with clear metrics and risk boundaries. Use these pilots to test technical viability, uncover organizational resistance, and generate internal proof points and champions.

5. Redefine Success Metrics

Classic project metrics may not apply in an adaptive world. Shift to metrics like system resilience (does the system adapt to change?), upgrade agility (how quickly can features be safely deployed?), user



experience personalization, and business outcome alignment. Make “adaptability” a key part of every new deployment and integration.

6. Prepare for Organizational Change

As MCP adoption grows, roles, responsibilities, and business models will evolve. Prepare your teams for more continuous collaboration – between product, support, service, and IT. Communicate openly about changing expectations and opportunities. Encourage curiosity, experimentation, and a willingness to embrace new ways of working and delivering value.



Appendix 2: Strategic Imperatives – The Future Impact of MCP

So, with the practical foundations laid, as and if organizations begin to experiment and embed MCP, and as the pace of change accelerates and the boundaries between AV, IT, workplace technology, and digital experience continue to blur, a much broader set of strategic questions and competitive realities come to the fore. Many of these stretch beyond MCP, and to the bigger picture of AI transformation enablement, but retaining an MCP lens, it's important to remember that MCP isn't just a matter of technical evolution – it's a strategic mindset leap for organizations and the entire AV ecosystem. So broadly the question leadership teams must explore are how are you:

1. Fostering Creativity and Use-Case Innovation

MCP enables organizations to reimagine what's possible. Instead of designing for static scenarios, AV and integration teams can create environments that adapt on-the-fly – meeting spaces that recognize participants and configure themselves, collaboration tools that surface context-relevant content, or support systems that pre-emptively resolve issues. The potential for differentiated user-driven experiences multiplies as MCP dissolves traditional integration barriers. For every use-case explored, there are a hundred more waiting to be unlocked by the organizations willing to think beyond legacy boundaries.

2. Leading in Industry Convergence

MCP lowers the barriers for collaboration with adjacent sectors – IT, real estate, facility management, security, and more. Traditional AV organizations have the opportunity to either lead in this convergence or risk being marginalized as new entrants, with mature IT or cloud-native capabilities, seize control of the user experience. Organizations that adapt quickly and focus on workflows, data, and orchestration – not just hardware – will thrive.

3. Cultivating Cultural and Organizational Agility

The move to context- and outcome-driven systems will stretch organizations culturally. Success will require new forms of collaboration, cross-disciplinary teams, and a willingness to blur traditional roles (project manager, programmer, system designer, end user). Training and leadership development must shift from “how to wire the system” to “how to design for adaptability and intent.”

4. Building a Risk-Tolerant, Learning Organization

Continuous improvement and adaptive integration are only possible when organizations become comfortable with experimentation and even occasional failure. MCP's flexibility means organizations can pilot new approaches, test value propositions, and iterate quickly – without risking catastrophic breakage. The future belongs to those willing to experiment early, learn fast, and operationalize new value as it emerges.



5. Driving Ecosystem-Wide Education

If MCP is to become foundational, its concepts must move from the domain of technologists to the language of business strategists, designers, and everyday users. Industry leaders must invest in education – internally and externally – so that everyone understands what MCP makes possible, what new roles it enables, and why the time to act is now.

Of course, as with any transformative shift, opportunities and risks evolve together. It's critical to consider both the practical and cultural challenges along the path to realizing MCP's potential.

