THE RISE OF AI AGENTS: A NEW ERA FOR MAINFRAME TRANSFORMATION



There is an old adage that says, “When the mainframe sneezes, the rest of the organization gets sick.” This remains true today as most business-critical applications depend on the mainframe. When problems do occur, the process to find and fix the problem is more of a process of elimination - is the problem a hardware or software problem, is it a batch or online issue, is it a database or storage problem - with the owner of each working to prove their part isn’t the one that caused the problem.

To make matters worse, a lot of monitoring solutions are siloed and only report on the part of the mainframe which they monitor. In the worst-case scenario, the issue stems from something buried deep in a database query written twenty years ago by someone who retired a decade ago. There are tools that can help, of course, but most of the time, the most powerful tool is a senior expert with institutional knowledge not found in any manual.

But what if that wasn't the only way? What if you could transform your mainframe with the help of a workforce of artificial intelligence (AI) agents that don't just assist, but collaborate, learn, reason, and act? What if you could simplify the complexity of mainframe systems by surrounding them with intelligent agents that work together like a team, operate autonomously, and understand your systems in plain language?

That’s the future of mainframe transformation—and for some, it’s already starting to take shape.

Fear of change

Let's back up for a moment. For decades, we've been caught in the same loop: legacy tools built for a world that no longer exists, supported by a shrinking pool of experts, all trying to keep mission-critical systems running without missing a beat. The result? Fear of change. Developers hesitate to modernize code they don't understand. Operators worry that tweaking one thing might break ten others. And system programmers spend hours chasing root causes through layers of logs, dumps, and outdated documentation.

That's the gap. It's not just a knowledge gap—it's a skills gap, a tooling gap, and a confidence gap.

That future is within reach — explore BMC's mainframe AI and GenAI solutions that put intelligent agents to work simplifying operations, accelerating code understanding, and enabling self-healing across the mainframe estate.

You've probably heard terms like "agentic AI agents," "autonomous agents," or "large language model (LLM)-based agents." Let's demystify this concept and, more importantly, put it into the context of how they could transform the way you develop, operate, and manage mainframe systems.

The coordinated workforce: Agentic AI agents

Think of a highly skilled workforce spread across different teams—each person brings a unique skill set to the table. Some specialize in troubleshooting, others in analysis, and a few in orchestrating the big picture. Individually, they contribute valuable insights. But when they work in sync, they become a powerful network of collaborators solving challenges together. Agentic AI agents that cooperate to solve problems need a reliable orchestration substrate — and the Amazon Bedrock integration allows enterprises to orchestrate Bedrock-powered AI agents as governed, scheduled steps in Control-M workflows alongside other IT and data processes. They can be homogenous (all similar) or heterogeneous (each one designed for a different task).

That vision of a coordinated AI workforce isn't just theoretical—it's already being explored in the industry. On a recent episode of The Modern Mainframe podcast, Jason English, an analyst at Intellyx, described it this way:

"It's like a team of mainframe specialists who can meet, collaborate, and decide amongst themselves how to solve a problem—without needing constant supervision. That's the power of agentic AI: coordinated intelligence that turns complexity into clear, actionable insight. It's not one agent that will be suitable for all aspects of AIOps, for instance, when dealing with system management and operations. We will see a collection of AI agents that will work in collaboration to proactively solve system issues."

In a mainframe environment, that might mean one agent continuously scans SMF data for anomalies while another watches CPU usage patterns and a third looks for changes in application behavior. A fourth agent acts as the coordinator, pulling together insights from the others to provide a clear picture of what's happening. Like bees sharing nectar, these agents share data and context in real time.

This isn't just monitoring—this is distributed intelligence. These agents don't just raise an alert, they work together to give you the why, the where, and sometimes even the how to fix it.

Self-driving mainframes: Autonomous agents

Now let's imagine a self-driving car. It perceives its surroundings, makes a plan, takes action, learns from what happens, and adjusts the next time it's used. That's how autonomous agents work.

In a mainframe transformation context, you could have an autonomous agent that notices an unusual spike in workload at 2 a.m. Instead of just logging it, the agent investigates. Was it a job rerun? A missed SLA? A rogue script? The agent then suggests a fix or even reschedules the job and adjusts thresholds based on learned patterns.

Or picture a developer saying, "Check all modules in this COBOL app for security vulnerabilities." The autonomous agent gets to work—it scans the code, applies pattern recognition, checks against known vulnerabilities, and generates a report with recommendations. Over time, it learns which types of vulnerabilities are common in your shop and becomes faster and more precise.

Autonomous agents don't just respond; they anticipate. And in doing so, they make your mainframe environment more adaptive and resilient.

The copilot you always wanted: LLM-based agents

Here's where things get even more exciting. LLM-based agents are powered by large language models and act like copilots. They reason, plan, use tools, and respond in natural language. Imagine the smartest assistant you've ever worked with—but instead of needing years of onboarding, it already understands your mainframe code.

That's exactly what tools like BMC AMI Assistant are designed to do. A developer opens a file and asks, "What does this code do?" The assistant reads the code, understands the logic, and responds with a clear explanation in natural language. An operator asks, "Why did this job fail last night?" and gets a contextual answer that pulls from logs, dumps, and performance data.

This is knowledge transfer at scale. The institutional knowledge that used to live in one system programmer's head is now codified, accessible, and explainable—instantly.

When They Work Together, You've Never Mainframed Like This

Now picture all three of these agent types working together:

- Agentic agents identify a problem: a pattern of job slowdowns happening across three applications.
- An autonomous agent investigates and discovers a DB2 performance issue due to an inefficient access path.
- An LLM-based agent explains the root cause, suggests a fix, and even offers to generate documentation or a test plan.
- A development feedback loop agent captures insights from the incident, integrates the fix into the CI/CD pipeline, and triggers a new build with updated code—closing the loop between operations and development.

Suddenly, your mainframe isn't a black box anymore—it's a transparent, collaborative environment where human and machine intelligence work side by side.

This is what we mean by "*mainframe, simplified*." This is what we mean when we say "*you've never*

mainframed like this."

Why the rise of AI agents matters now

The rise of AI agents isn't just a trend—it's a shift in how we think about mainframe transformation.

Mainframe transformation isn't just about rewriting code or lifting and shifting workloads. It's about removing the barriers that have historically held us back—barriers like institutional knowledge locked away in a retiring workforce, brittle systems no one wants to touch, and tools that were never designed for agility.

When AI agents become part of your daily workflow, something remarkable happens. Developers stop fearing the code they don't understand and start building with confidence. Operators no longer spend all night piecing together clues—they resolve issues with clarity and speed. And system programmers finally get to share their expertise in a way that's scalable and accessible to the next generation.

These aren't just tools—they're trusted teammates. They learn. They adapt. And they speak your language.

That fear of change that's been lingering in the background, hindering modernization efforts? It begins to fade. In its place, there's a new mindset: one that sees the mainframe not as a system you're stuck with, but as a system with which you can *evolve*.

It's not about replacing people—it's about giving people the superpowers they need to move faster, solve smarter, and transform with confidence.

And in that shift, the mainframe becomes more than a foundation for critical business systems. It becomes a platform for what's next.

That's what could be. And for more and more organizations—it's already beginning.

So, the next time someone asks you how you plan to modernize your mainframe, you might just say, "We're letting the agents handle it." Not because you're stepping away, but because you're stepping forward into a new era where AI doesn't replace expertise—it scales it. And with that, the mainframe becomes not just a system of record, but a platform of innovation.

This is the rise of AI agents—a new chapter in how we simplify the complex, bridge generations of expertise, and make the mainframe more intuitive, intelligent, and indispensable than ever before.

That's what could be. And honestly, it's already starting to be.

Learn more about AI Agents, LLMs, generative AI (GenAI), hybrid AI and more in the Modern Mainframe podcast, "The Future is Hybrid: 2025 Predictions for GenAI," featuring Intellyx analyst Jason English.