One Codebase, Many Platforms: Why the AI Era Demands a Unified BMC

Insights from OpenBMC community – An industry point of view paper
By Earle Philhower, Marketing Manager for MegaRAC® Manageability Solutions, AMI
July, 2026

For most of my career in data center infrastructure, the baseboard management controller (BMC) was the quietest part of the rack — essential, but rarely talked about. It booted the box, watched the fans, and stayed out of the way. Engineers treated it as plumbing. Buyers rarely asked about it. The firmware that ran it was a footnote in a procurement document.

That era is over. The BMC has become the control plane and the security boundary for AI infrastructure. It carries telemetry off every node, manages lifecycle, power and thermal behavior, enforces policy, and serves as the integration point between the physical platform and the orchestration software running above it. When a cluster spans tens of thousands of accelerators, the firmware layer is where fleet-scale visibility either exists or does not. It is where the security posture is set. It is where the difference between a managed system and an unmanaged one is decided.

The companies building the largest AI systems already understand this. At the inaugural AMI and Meta OpenBMC Meetup, hosted at Meta's Menlo Park campus in May 2026, NVIDIA's OpenBMC maintainer Ed Tanous put the dependency plainly:

> *"NVIDIA uses the hell out of OpenBMC."*
> \- Ed Tanous, OpenBMC Maintainer, NVIDIA

That is not a hobbyist statement. It is a signal that open firmware now sits underneath some of the most valuable compute on earth.

Sanjoy Maity, SVP and AMI Business Unit Leader at AMI, a Lattice Company, has a useful name for what operators are up against at this layer. He calls it a five-headed hydra. Each head carries its own cost: a direct financial hit, a time-to-market penalty, and a steady climb in operational complexity.

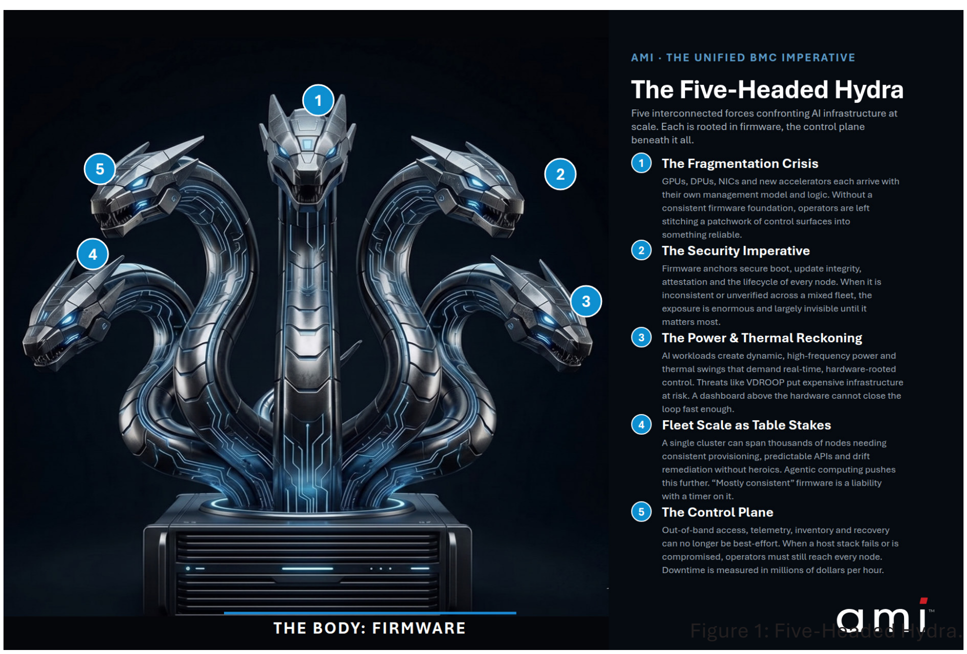

Figure 1: Five-Headed Hydra.

Every one of those costs traces back to the firmware layer. This paper takes the hydra one head at a time, across fragmentation, security, power and thermal, fleet scale, and the control plane, listening to the engineers who manage them at the world's largest operators and following the money. The industry has agreed, in practice, that open firmware is the foundation for managing AI infrastructure. It has not yet agreed on how to keep that foundation coherent as the money and the hardware scale past anything the field has seen.

The Investment Imperative

Let's start with the size of the build, since the numbers govern every engineering decision underneath them. McKinsey projects that data centers will require roughly $6.7 trillion in cumulative capital expenditure by 2030, with $5.2 trillion of that tied to AI-specific infrastructure. The AI data center market alone is forecast to grow from $471.6 billion in 2026 to $2.02 trillion by 2032, a compound annual growth rate of 27.5%, according to MarketsandMarkets. The top five hyperscalers are expected to spend more than $600 billion on infrastructure in 2026, a 36% jump year over year, and analysts at CreditSights and MUFG estimate roughly 75% of that, about $450 billion, is aimed at AI.

The physical envelope is climbing just as fast. AI racks now draw 100 to 200 kilowatts and more, individual clusters routinely pass 100,000 GPUs, and the International Energy Agency projects data center electricity demand will cross 1,000 terawatt-hours in 2026. Capital flowing into electrical and cooling systems for AI data centers is on track to reach about $220 billion a year through 2030, more than triple the 2024 level.

> *"The greatest buildout of compute infrastructure in our lifetimes is underway. And there is no end in sight for us with trillions of dollars flowing for new deployments within the next few years alone."*
> - Zachary Bobroff, VP Product Management, AMI, OCP EMEA Summit, April 2026

Here is the part that gets too little attention. Every dollar of that spend buys hardware that firmware has to manage. The five heads that follow are the places where management cost is created or controlled. Each one is an engineering problem with a price tag attached.

Head One, The Fragmentation Crisis: Forking Was a Feature, Until It Became a Bill

The economics only make sense if you grant that forking was rational. OpenBMC™, a Linux Foundation project, gives the industry a common BMC firmware stack that works across heterogeneous systems. It succeeded enough to become essential and fragmented enough to become expensive. Both are true at once, and the people closest to the project say so directly.

> *"I don't see (forks) as a problem. I think if people are forking it and modifying it, it means they're using it, and they're using it in ways that the original authors didn't. The number of forks is more a sign of success than a sign of failure."*
> - Patrick Williams,
> Software Engineer, Meta

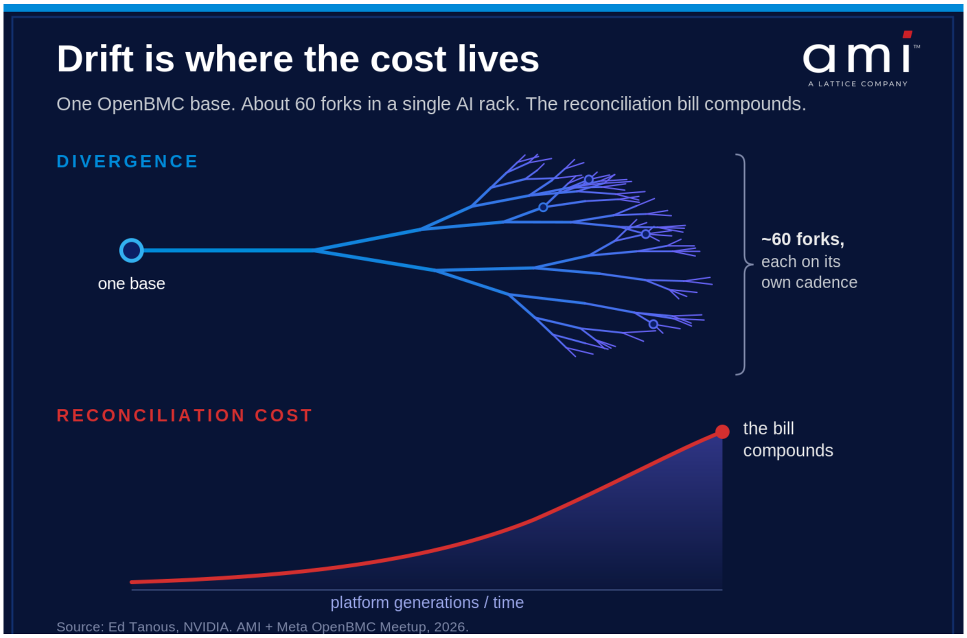

Figure 2: Diverging codebase versions drives increased reconciliation cost.

Every fork is a vendor that found the base code valuable enough to invest in extending. Each fork tracks a specific hardware vintage or a single vendor's platform, and they drift apart over time. Drift is where the cost lives. Every divergence becomes work someone has to reconcile later. Ed Tanous described an NVIDIA GB200 system, noting there are "somewhere in the range of about 60 different OpenBMC implementations running in various capacities" in that single rack. Sixty different codebases being built and maintained, in one rack, each on its own cadence. That is the fork story in one image, and it is the seam where the bill starts to compound.

> *"There are over 15 relevant forks out there today in the OpenBMC community, and most of these tend to be driven by either SoC vendors for BMCs or host silicon from different hardware vendors."*
> - Zachary Bobroff, VP Product Management, AMI

Head Two, The Security Imperative: A Larger Attack Surface, Built One Fork at a Time

Firmware does more than boot a server. In a multi-tenant AI data center it anchors secure boot, governs update integrity, enables attestation, and manages the lifecycle of every node. The attack surface rises with each fork, and the value of what sits behind it rises faster. During the AMI and Meta OpenBMC Meetup, Nuvoton's Hila Miranda-Kuzy walked through an integrated root of trust built into the BMC, and her framing showed why the firmware layer is the place to set the security floor.

> *"The first motivation for integrating the root of trust with the BMC into the same solution is to be able to avoid any kind of attacks that are related to substitution."*
> -Hila Miranda-Kuzy, BMC SW Application Engineer, Nuvoton

The risk of leaving that floor uneven is now empirical. A 2026 study presented at the IEEE International Conference on Information Networking found that vendor-specific BMC implementations carry divergent vulnerability patterns, a direct consequence of isolated development rather than shared, community-hardened code. Different forks, different holes. The mitigation is not secrecy. As Jessie Frazelle argued in ACM Queue, open-source firmware "can help bring computing to a more secure place by making the actions of firmware more visible and less likely to do harm." When a zero-day can surface and spread in minutes, transparency and fast, shared patching beat obscurity.

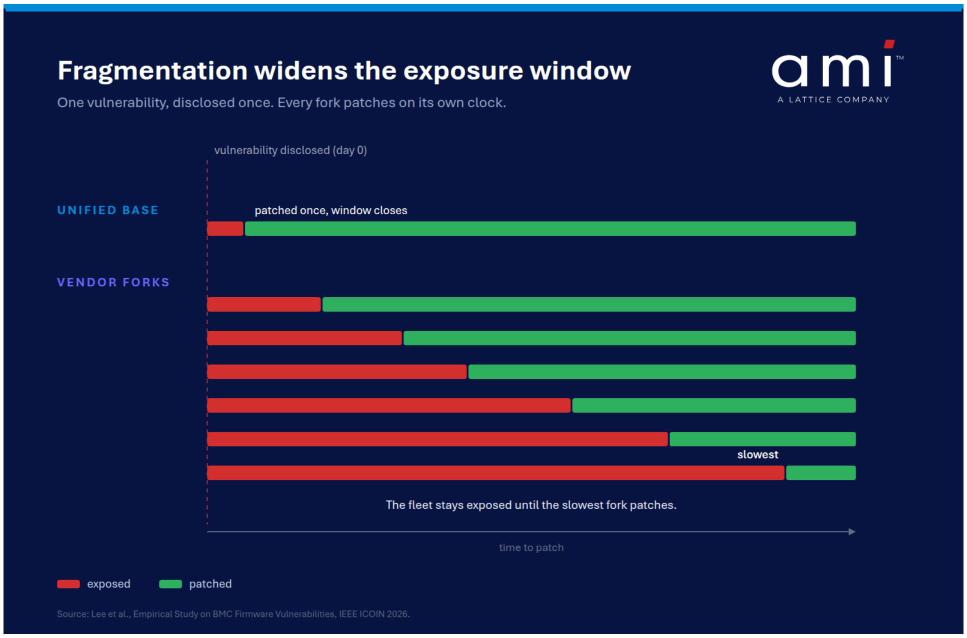

Figure 3: Each codebase fork increases the time to patch vulnerabilities.

The economics are blunt. A breach in an AI data center is not a line item, it is a board-level event, and exposure that is invisible across a mixed fleet is the most expensive kind. Fragmentation forces every vendor to harden its own branch on its own schedule, which multiplies the work and widens the window. A shared base avoids that completely. Hardening done once, upstream, defends every operator that builds on it. Security is the clearest case where a common foundation turns duplicated cost into shared protection.

Head Three, The Power and Thermal Reckoning: When Firmware Runs the Cooling

The steady, predictable power baselines that data center teams relied on for years are gone. AI workloads create dynamic, high-frequency power and thermal swings that demand real-time control rooted in hardware, and the firmware layer is where that control now lives. At the OpenBMC Meetup, Patrick Williams made the point by describing how far the BMC's job has already spread. Walking through a diagram of a Meta rack, he pointed at the cooling hardware and what now manages it, saying:

> *"That's an air assisted cooling device. And that actually has a BMC like device in it as well to control pumps and detect leaks."*
> -Patrick Williams, Software Engineer, Meta

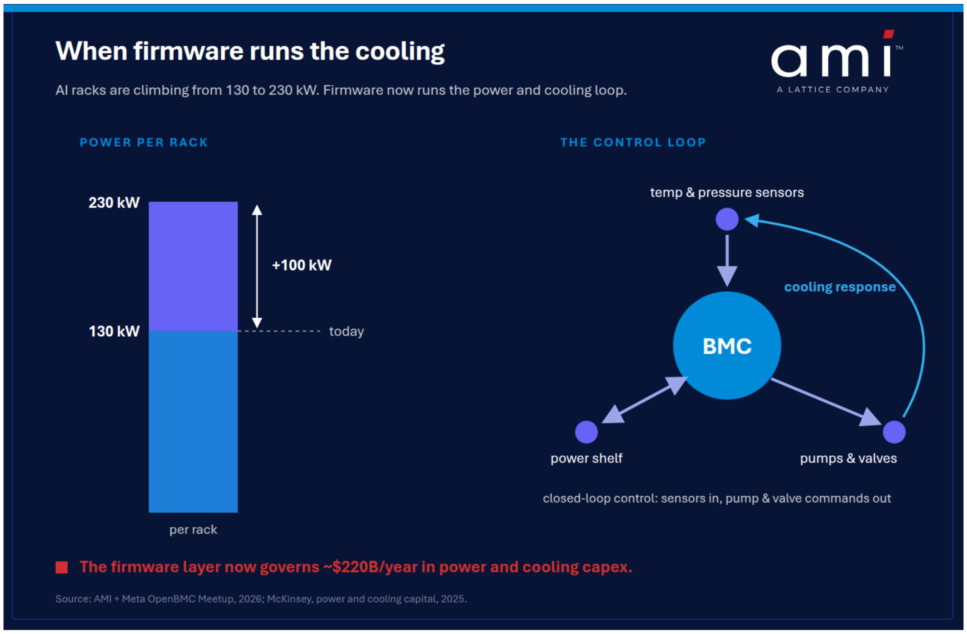

Figure 4: Firmware now runs the power and cooling loop as AI racks climb from 130 to 230 kW.

As rack power climbs from 130 kilowatts toward 230 kilowatts, cooling distribution units move from in-rack to in-row, power shelves move to their own rack, and the BMC takes on new roles. In power shelves it becomes the Power Management Controller, talking to supplies over Modbus and PMBus. In a cooling distribution unit, it runs a closed-loop flow control that holds coolant at an optimal rate based on temperature and pressure sensors, and detects leaks. Firmware is no longer adjacent to the power and cooling plant. It is the thing running it.

The cost stakes here are the highest in the building. With roughly $220 billion a year flowing into power and cooling for AI data centers, the firmware that governs those systems decides whether the most expensive physical infrastructure runs at the edge of its envelope safely or strands capacity to stay clear of failure. A fragmented power and thermal control layer cannot optimize across a fleet, and a single thermal event in a high-density rack can destroy hardware worth more than the firmware program that was supposed to protect it. This is the head of the hydra that turns a software-consistency argument into a capital-protection argument.

Head Four, Fleet Scale Is Now Table Stakes: Small Inconsistencies, Millions of Nodes

What looks manageable in one rack becomes a structural burden across a fleet. Microsoft's Thirupathaiah Annapureddy, who leads Azure control plane firmware engineering, described the surface area his team manages:

> *"We're kind of deploying millions of nodes, and most of the nodes have either one or multiple BMCs."*
> - Thirupathaiah Annapureddy, Partner Software Engineering Manager, Microsoft

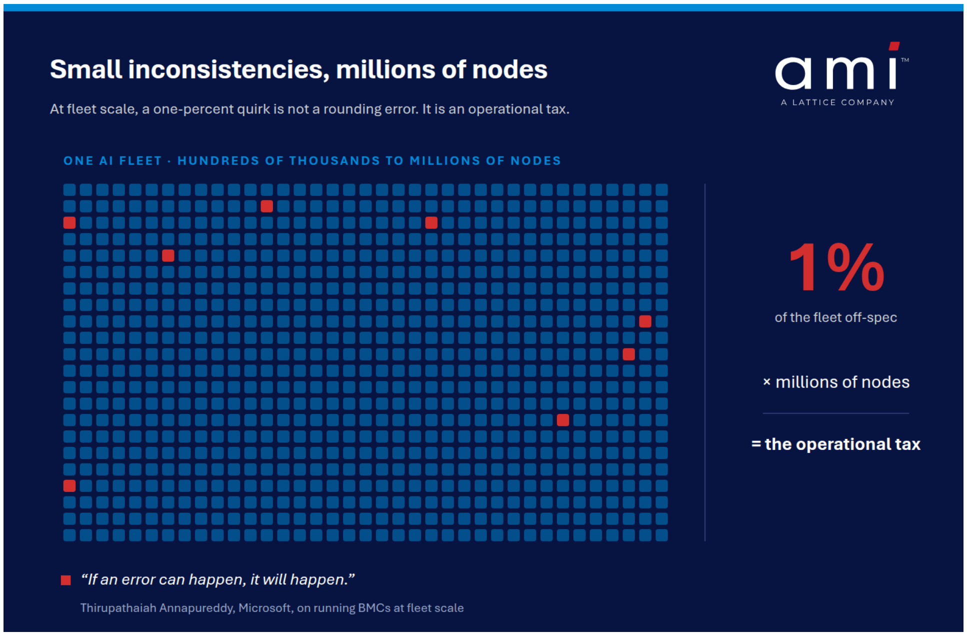

Figure 5: At fleet scales, a one-percent error rate is the operational tax of operating millions of nodes.

Spread across hundreds of platforms and SKUs, mixing Intel, AMD, NVIDIA, and Microsoft's own silicon, that surface area turns small inconsistencies into operational risk. His read on life at that scale was blunt: "If an error can happen, it will happen." Operators absorb the integration work that fragmentation creates. Williams described the daily reality for a systems integrator running mixed hardware:

> *"Do I take from ASPEED's fork this bit, and from NVIDIA's fork this bit, and from AMD's fork this bit, and how do I put them all together into something that I can release and manage and deploy to my fleet?"*
> - Patrick Williams, Software Engineer, Meta

Meta answers that question with a small, disciplined team that refuses to build on any single vendor's fork and insists on integrating vendor components into one standard tree. Not every operator has the engineering depth to hold that line. The pattern is not unique to firmware. A Rimini Street and Censuswide survey of more than 600 CIOs and CTOs found that nearly 75% consider managing a growing number of vendors and products an inadequate model for their IT needs, with about a third pointing to mismatched vendor processes and excessive management effort. BMC fragmentation plants that same friction at the lowest software layer in the stack, where it is hardest to see and most expensive to unwind. At a million nodes, a 1% inconsistency is not a rounding error. It is the operational tax that scales with every server added.

Head Five, The Control Plane Is Mission Critical: When the Host Goes Dark

Out-of-band access, telemetry, inventory, and recovery cannot be best-effort capabilities. When a host stack goes down or gets compromised, and at scale it will, operators need to see and reach every node in the fleet. An Intel engineer brought the room back to first principles on what the BMC is for:

> *"Going back to our DNA, what BMC is all about is lights out management. When everything else is not working, BMC has to work... Because it is the one that's going to prevent and recover."*
> - Nirav Shah, Software System Architect, Intel

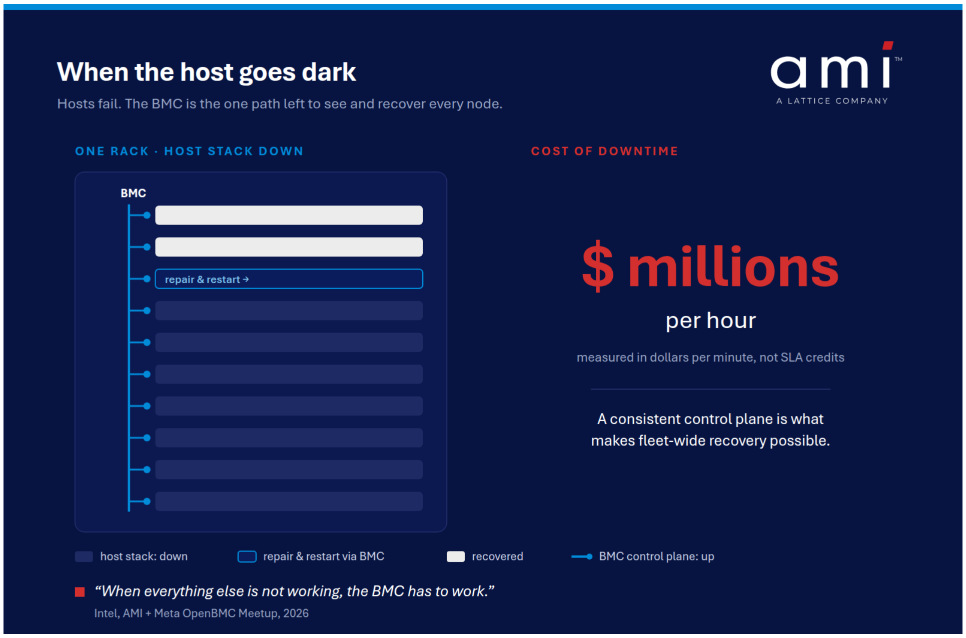

Figure 6: The BMC is the one path left to see and recover every node when the host goes dark.

Recovery runs on that same foundation. Microsoft's team described using the BMC as the instrument that brings failed nodes back into service.

> *"We use BMCs, for example, to bring the nodes back to productization. We will have to first figure out why a node or a server is not available. So for that, we need telemetry from the BMCs."*
> - Thirupathaiah Annapureddy, Partner Software Engineering Manager, Microsoft

With AI infrastructure costs at hundreds of billions of dollars per year, downtime is not measured in SLA credits, it is measured in millions of dollars per hour. A control plane that behaves differently across vendors fails in exactly the moment uniform behavior matters most, when an operator is trying to reach thousands of nodes at once under pressure. Consistency at this layer is not a tidiness preference. It is the difference between a fast, fleet-wide recovery and an outage that bleeds money by the minute.

The Unified Model: A Distribution, Not a Fork

Five heads, one body. The reason the hydra framing holds is that every head traces back to the same firmware layer, which means a single discipline can cut all five at once. The fix is not another fork that promises to end forking. It is a shared, open, production-ready codebase that supports many silicon vendors and platform generations from one tree. Two practices make a distribution work rather than splinter. The first is release and versioning discipline, which the OpenBMC Project itself acknowledges is unfinished work. Williams was candid about the gap.

Most of the industry cannot operate that way, and Williams was clear that serving them is unsolved work for the community.

> *"I think it's something that the community needs to do a better job solving, is how do we make a release of some sort and provide support for that in some meaningful way."*
> - Patrick Williams, Software Engineer, Meta

The second practice is upstream-first development paired with a stable downstream path. AMI's MegaRAC® OneTree Community Edition is one model of that approach, a hardware-agnostic distribution built on the OpenBMC codebase and offered through the OCP Marketplace.

> *"We're trying to kind of herd these forks all back into something that's common. We're not looking to create another fork here."*
> - Zachary Bobroff, VP Product Management, AMI

Differentiation still lives on top of the shared base, which is the point. The integrated root of trust that Nuvoton demonstrated is exactly the kind of value a vendor should build above a common foundation, not the kind of divergence that fractures it. The same model extends past the server. OpenBMC now runs on rack managers, power shelves, cooling units, switches, and co-packaged optics, and from one codebase a team can derive firmware for all of them. Celestica's Avinash Natarajan came at the problem from the operator side:

> *"I'm not a true blue OpenBMC guy. I come more from a data networking background, but I got into building an orchestration software for managing a rack."*
> - Avinash Natarajan, Senior Principal, Celestica

The people who operate this infrastructure want one consistent surface to manage it through. A distribution earns trust by upstreaming what the project will accept and reserving its value for the layers above the shared base. The analogy to Linux is exact. The kernel did not fail because Ubuntu and Fedora built on top of it. It succeeded because distributions created a stable, opinionated path to production that organizations could adopt without forking the kernel itself. The BMC ecosystem needs the same discipline.

Governance Is the Gating Problem

Convergence is now a coordination problem, not a capability problem. The technical pieces exist. Williams noted during the Meetup that he had just tagged OpenBMC 3.0, the project's first release in more than a year and a signal that the community is converging on shared release practices and security work. The question is whether the industry will manage fragmentation deliberately or let it compound as the build accelerates.

This is where industry posture matters more than any single product. The ecosystem does not need another vendor that claims openness and steers customers toward lock-in. It needs vendors developing upstream first, contributors working in public, and operators willing to push their silicon partners toward shared standards. Linux Foundation research on open-source fragmentation makes the structural case plainly: fragmentation, left unmitigated, raises costs and complexity for everyone and reduces the impact of a large community collaborating around a shared platform.

The math is not subtle. Trillions are flowing into AI infrastructure on the assumption that it can be managed at scale. That assumption holds only if the control plane converges. Apply even a conservative 15% to 20% management overhead to a market headed toward $2 trillion a year, and the waste embedded in a fragmented control plane runs to the hundreds of billions annually. Research on AI data center lifecycle management points to total cost of ownership reductions of 15 to 40% from cross-stage optimization, and a separate analysis found the large majority of organizations report efficiency losses when they run across fragmented, non-optimized stacks. Cut one head and the others remain. A unified, open foundation is the only cut that takes all five at once.

That foundation already has a starting point. MegaRAC OneTree CE puts a common, upstream-first BMC codebase in the open, where operators and silicon vendors build on shared standards instead of maintaining sixty private forks. The path to a coherent control plane runs through open principles and broad distribution: develop in public, contribute upstream first, and push silicon partners toward one base that everyone extends rather than replaces. The money is already committed. The question is whether the industry manages it on a foundation it shares or one it keeps rebuilding apart.

> *"The OpenBMC community needs more vendors developing upstream first, more contributors working in public, and more operators willing to push their silicon partners toward shared standards."*
> - Patrick Williams (Meta) and Brian Vandecoevering (AMI), closing remarks, inaugural OpenBMC Meetup, May 2026

Figures and illustrations are conceptual representations of topics discussed at the inaugural AMI–Meta OpenBMC Meetup, 2026.

References

1. McKinsey & Company, "The cost of compute power: A $7 trillion race" and "Who's funding the AI data center boom?" (2025).
2. MarketsandMarkets, "AI Data Center Market worth $2,023.52 billion by 2032" (March 2026).
3. CreditSights / MUFG Americas, hyperscaler capex 2026 estimates (2025).
4. International Energy Agency, data center electricity demand outlook, via DCD (2024 to 2026).
5. Rimini Street / Censuswide, CIO and CTO multi-vendor support survey (2023).
6. Lee et al., "Empirical Study on BMC Firmware Vulnerabilities," IEEE ICOIN 2026, DOI: 10.1109/ICOIN68469.2026.11480507.
7. Jessie Frazelle, "Open-source Firmware," ACM Queue 17(3), 2019.
8. "Rearchitecting Datacenter Lifecycle for AI: A TCO-Driven Framework," arXiv (2025).
9. Linux Foundation Research, "Collaboration, Alignment Key to Reducing Open Source Fragmentation" (January 2023).
10. Sanjoy Maity, "The 5-Headed Hydra of AI Data Center Control," AMI (April 2026).
11. AMI and Meta inaugural OpenBMC Meetup, Menlo Park, May 2026. All speaker quotes drawn from session recordings: Patrick Williams (Meta), Zachary Bobroff (AMI), Ed Tanous (NVIDIA), Thirupathaiah Annapureddy (Microsoft), Nirav Shah and Jason Bills (Intel), Hila Miranda-Kuzy (Nuvoton), Avinash Natarajan (Celestica), and the Session 10 rack-management panel (AMI, Lattice, NXP).
12. OpenBMC 2.18 and 3.0 release notes, OpenBMC project (2025 to 2026).