

SEIZING QUANTUM'S EDGE

HPC-QUANTUM-AI : SHAPING THE NEXT COMPUTE ERA



ALICE & BOB



HYPERION RESEARCH



FOREWORD:

In Seizing Quantum's Edge (2025), we made the case for preparing now, and showed which applications early fault-tolerant quantum computing would reach first. The case has held.

A year on, hardware has moved faster than most expected. What has not kept pace is everything wrapped around it: the compiler that translates a scientist's intent, the scheduler that queues the job, the decoder chasing errors off the chip.

That gap is where this report goes. Fifteen HPC directors, quantum leads, and engineers told us what is actually breaking. Their answers converge on three moves: navigate before you commit, unify before fragmentation sets, co-design before the machines arrive.

Readiness does not ship with the hardware. It is built in the years before it lands.

METHODOLOGY:

This report was written to pair practitioners' experience with our perspective and actionable recommendations.

The findings draw on interviews conducted in Q2 2026 with fifteen directors, quantum program leads, and engineers across Europe, APAC, and the United States, complemented by a review of the published literature.

WE THANK

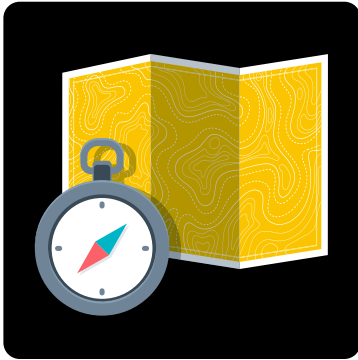


the experts at Argonne National Laboratory, Lawrence Berkeley National Laboratory / NERSC, Oak Ridge National Laboratory, AIST / G-QuAT, LINKS Foundation, Pawsey Supercomputing Centre, Munich Quantum Valley / Munich Quantum Software Company, CEA, OCP, RIKEN, Marqov, SURF, bp, Dell, and GE Vernova for sharing their insights.

Interviewees spoke anonymously; quotes are identified by role only.

TAKEAWAYS

FOR THE HPC-QUANTUM-AI COMMUNITY



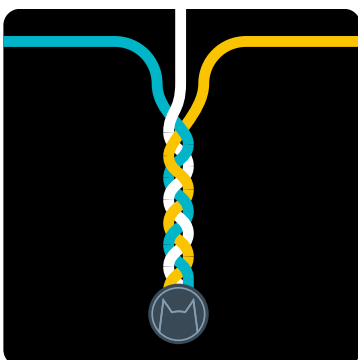
1 NAVIGATE

No one has yet proven how tightly quantum must couple to classical compute, and the answer can't be bought, only measured. The supercomputing centers running testbeds and latency experiments today are the ones who will commit capital on evidence rather than on a hypothesis.



2 UNIFY

Every vendor stack integrated by hand is months lost to plumbing. The way out runs through interoperability: common ground on the layers that don't depend on qubit modality, declaring resources, submitting jobs, built openly, so the whole community moves together rather than apart.



3 CO-DESIGN

FTQC will not be a drop-in accelerator. The gains that matter are surfacing at the seams between disciplines, where domain scientists, HPC and AI experts, and quantum vendors sit around the same problem. The workflows won't be ready when the machines land unless that work starts now.

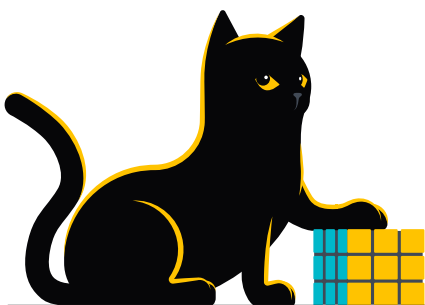




Table of contents

• Mind the maturity gap: the software stack will make or break quantum in HPC	4
1• Navigate the HPC-Quantum-AI design space	6
2• Unify the software stack, interoperability is the way through	9
3• Co-design the HPC-Quantum-AI application workflows	11
• Expertise is built in-house: The case for on-premise deployment	14

MIND THE MATURITY GAP : the software stack will make or break HPC-Quantum-AI integration

PHYSICISTS HAVE THE HARDWARE DOWN, BUT HPC MUST BUILD THE SOFTWARE STACK UP.

The HPC center director lies awake worrying. Not over qubits: those will arrive within a few years, packaged and humming to physicists' tunes. It's everything wrapped around them: the compiler that translates a scientist's intent, the scheduler that slots a job into

the queue, the monitors that catch a fault mid-run, the error correction chasing a torrent of data off the chip. The software stack. The hidden layers that turn a raw quantum processor into a usable scientific computing platform.

Building blocks in the stack for a useful fault-tolerant quantum computer

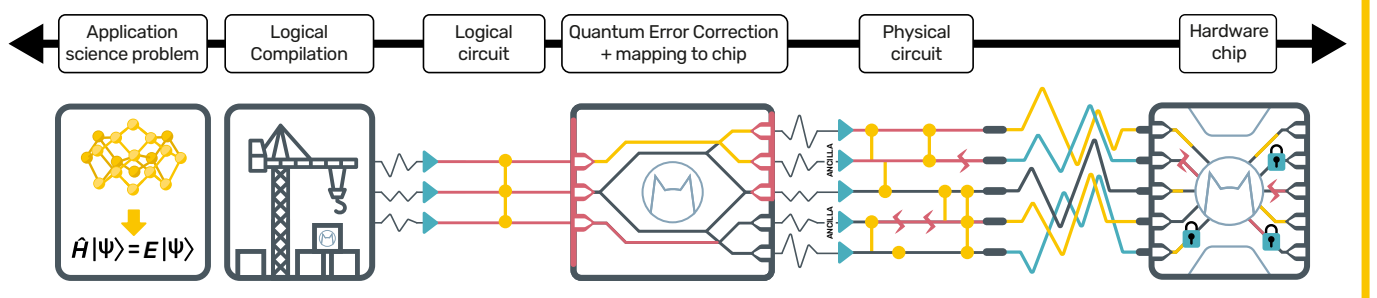


Figure 1 - Illustrative and simplified view of the stack, from a user's science question to the output of the quantum computer. The problem is expressed as a quantum circuit, compiled into physical gates protected by quantum error correction, then executed on the hardware chip.

Right now, those layers lag. Hardware sprints; software meanders. And the leaders steering the world's supercomputing centers know precisely what happens if the gap fails to close in time.

One director spells it out: **"If I get errors in my compiler and I've got a deadline, ... [this] could kill quantum, because if the software stack is not ready, people will turn their nose up at quantum computing."**

Trust erodes faster than it rebuilds. Let quantum stumble in front of a community that already runs reproducible, end-to-end workflows on demand, and the skeptics win the argument. The setback gets measured in years.

A quantum lead rips off the band-aid without flinching and tells it straight:

"RIGHT NOW EVERYTHING SEEMS DUCT TAPED."



Quantum lead at a HPC Center.

Speaking of the then promising trend (c.2008) toward multicore architecture, Donald Knuth said *"the HP/Intel Itanium approach [...] was supposed to be so terrific, until it turned out that the wished-for compilers were basically impossible to write."*

— Knuth, Donald E. (April 25, 2008). "Interview with Donald Knuth". InformIT. Archived from the original on February 23, 2021. Retrieved August 10, 2026.

So the qubits edge closer to fault tolerance while the infrastructure trails behind. Whether quantum can compute will not be decided by qubit quality alone, but by who builds the stack together. That work now needs to accelerate. ■

Critical FTQC abstraction layers and their classical interfaces are missing

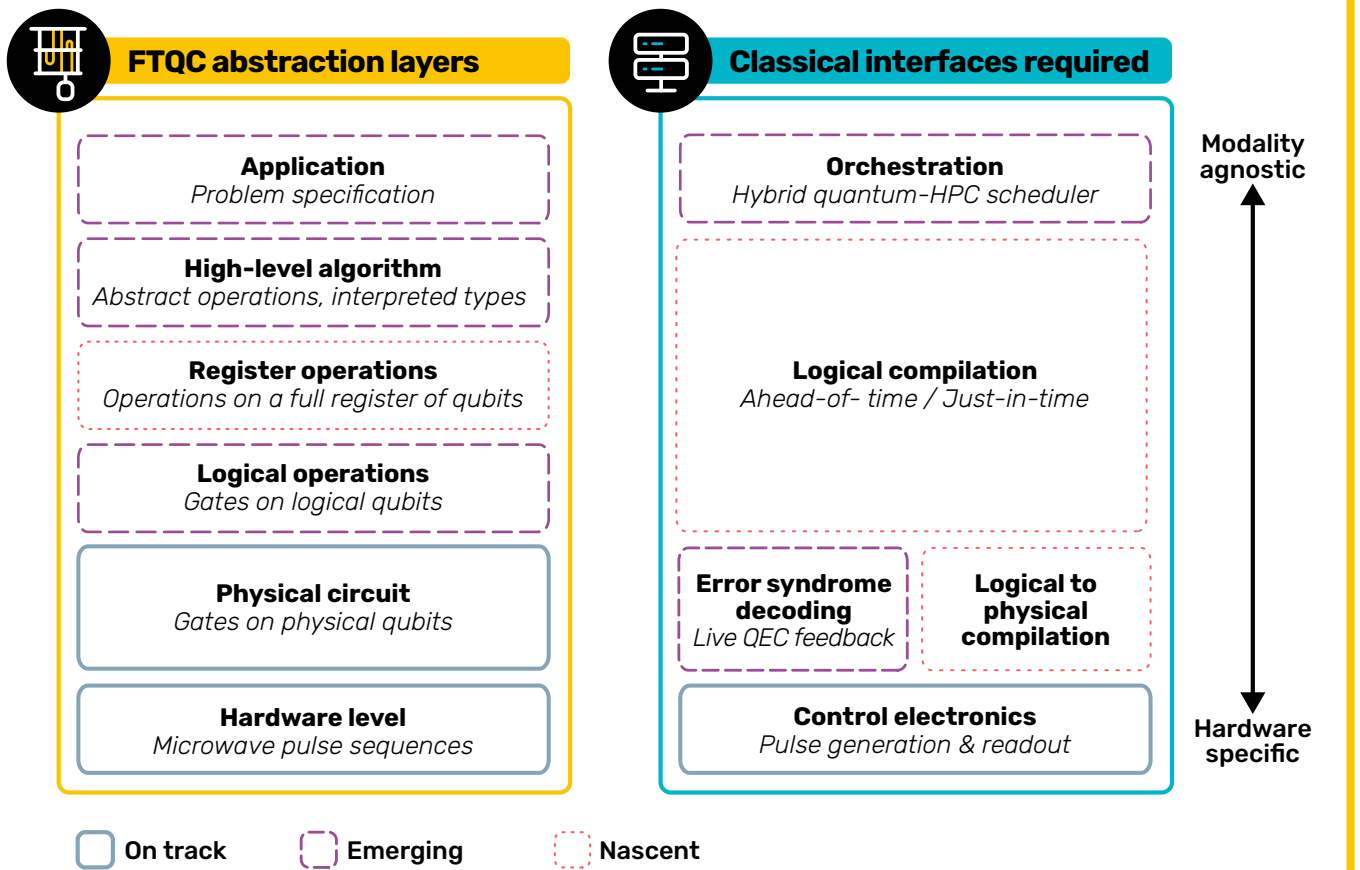
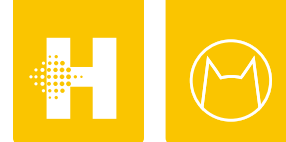


Figure 2 – Maturity is not homogeneous in implementation across different layers of the quantum-HPC stack.



1. NAVIGATE the HPC-Quantum-AI design space: the paths to integration are still open

NO ONE HAS THE ANSWER YET - ONLY EXPERIMENTS SHOW THE WAY.

A signal leaves a quantum chip. The signal carries one bit per qubit, streaming out at a million cycles a second, and somewhere downstream a classical machine has to catch it, decipher it, and respond before the next cycle arrives. Miss that window, and the backlog grows. The backlog compounds. The processor stalls. For the people who are advancing the world's supercomputing centers, that microsecond is where the next decade of architecture gets decided, and right now, nobody knows exactly how it will play out.

They know the stakes. HPC center directors have spent careers coaxing performance out of machines like Fugaku, which recently lent 6,400 of its 152,000 nodes to an IBM quantum processor to compute the excited-state energies of biomolecules made up of more than 12,000 atoms^[1]. They have seen quantum move, as one team at a leading HPC center put it, from "Can a quantum processor exist?" to "How can I practically build and operate one?" The same centers are meanwhile rebuilding themselves around AI, so the two transitions are colliding in the same machine hall. Now vendors are arriving with roadmaps, reference architectures, and a pointed question: how tightly should you integrate a quantum processor to your classical AI supercomputer?

Here is the uncomfortable truth. The design space is still open. Classical control at the rack level has made a lot of progress; the open opportunity is the next tier up, where quantum meets full-scale HPC and AI machines. No one has proven where quantum and classical compute must fuse and where they can stay comfortably apart, which means the leaders res-

possible for evaluating this technology cannot buy their way to an answer. They have to build small, wire up testbeds, and run the latency experiments themselves, before committing capital to a full-scale system built on a hypothesis.

Consider the two camps.

At one extreme lies a comical yet apt scene: a carrier pigeon lofting an SD card. Quantum computers excel at "small data, big compute" problems. Shor's algorithm factoring a 2048-bit RSA integer takes hours to days but moves only a few kilobytes, a common SD card would hold it with room to spare^[2].



The pigeon lark lands because it exposes a valid point: for many hybrid quantum / classical workflows, loose integration and batch scheduling are enough. Tight, low-latency plumbing would be effort spent solving a problem you do not have.

At the other extreme sits the syndrome stream, and it is no joke. Fault tolerance demands quantum error correction, and it is punishing. ▶▶

How tightly to integrate classical to quantum, and where AI belongs, comes down to: how much latency can the task tolerate?

[1] S. Seelam et al., "Reference Architecture of a Quantum-Centric Supercomputer," Mar. 12, 2026, doi: 10.48550/arXiv.2603.10970.

[2] C. Gidney, "How to factor 2048 bit RSA integers with less than a million noisy qubits," May 21, 2025, doi: 10.48550/arXiv.2505.15917.

► For surface-code architectures, estimates run as high as a thousand fragile physical qubits to protect a single logical one, about half of them holding the state you care about, half watching for phase and bit-flip errors. For fast superconducting qubits, that watching produces a torrent of so-called "syndrome" data at up to a million cycles per second, and a classical decoder must keep pace while sustaining roughly a terabit per second of throughput and a petaflop of

compute for the scientifically relevant quantum computers coming in a few years^[3]. Decoding decisions may need to be returned within roughly ten microseconds to keep the quantum computer operating at full speed. That is barely enough time for processing huddled close to the readout hardware. Push a thread of control out to networked accelerators, and added latency slows the overall computation.

Where live AI fits in the quantum stack

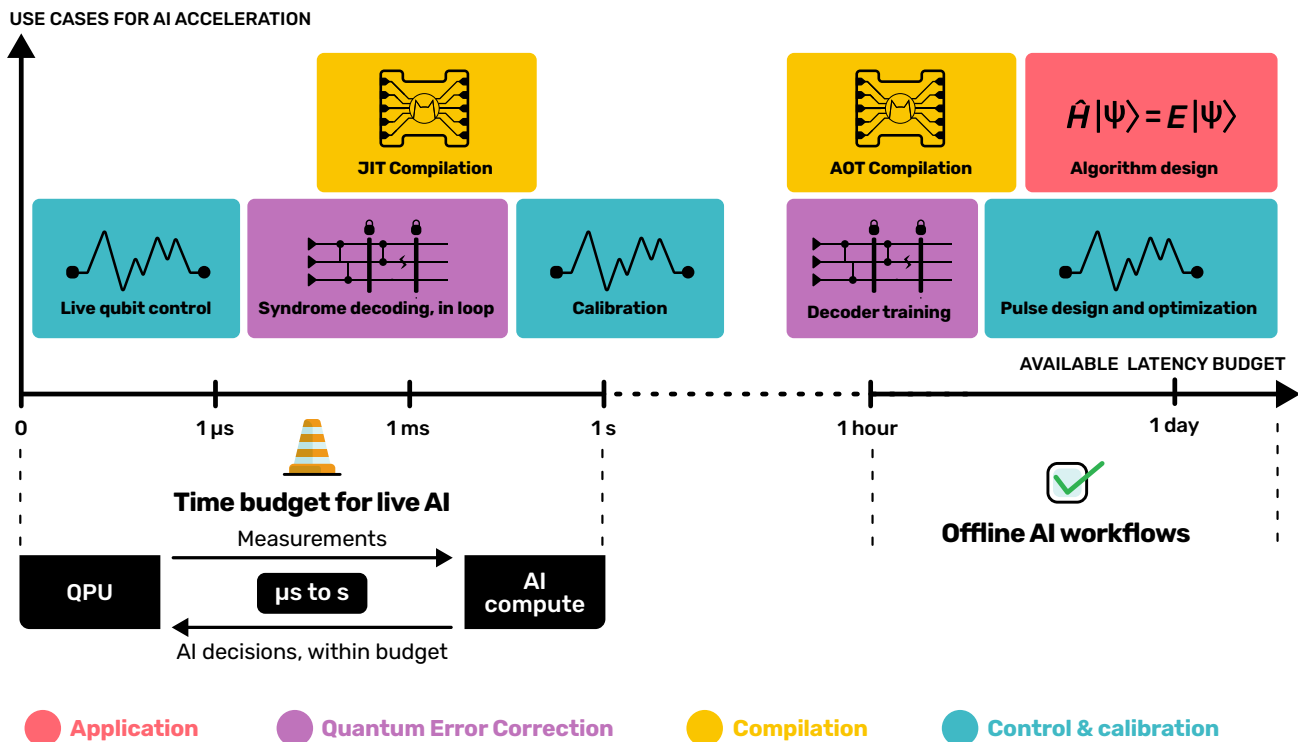


Figure 3 – AI workloads across the fault-tolerant stack, positioned by the latency budget each must meet. Where budgets stretch to hours or days, AI has already proven itself; where they shrink to microseconds, only closed-loop testbeds between QPU and AI compute can show whether it holds at scale.

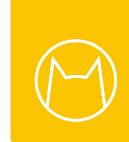
Ahead of time (AOT) compilation has no clear real-time constraint yet. There are trade-offs to be made between compile-time and run-time overheads that are not yet fully understood. AI systems can explore the large design space of fault-tolerant compilation (error correction, qubit routing, space-time trade-offs) offline, with GPU-powered simulators validating these schemes. But some just-in-time (JIT) compilation schemes can still require tighter coupling with classical compute resources, though their development and testing remain to be done.

Decoding spans both ends of the latency spectrum. Training an AI decoder can be done with loose coupling, running them cannot.

Calibration, however, has another set of considerations. As described in a recent blog post by Alice & Bob's Kevin Kissell, copies of the measurement stream could feed AI models that identify drifts in parallel and update the parameters of the real-time firmware, without touching the microsecond loop^[4]. ►

[3] D. Camps, E. Rrapaj, K. Klymko, B. Austin, and N. J. Wright, "Evaluation of the Classical Hardware Requirements for Large-Scale Quantum Computations," in *ISC High Performance 2024 Research Paper Proceedings (39th International Conference)*, Hamburg, Germany: IEEE, May 2024, pp. 1–12. doi: 10.23919/ISC.2024.10528937.

[4] K. Kissell, "Decoupling AI for Quantum Control and Calibration," Jun. 2026, Available: <https://alice-bob.com/blog/decoupling-ai-for-quantum-control-and-calibration/>



➤ This is where AI comes in. NVIDIA's NVQLink – an open architecture for tightly coupling QPUs with GPU-accelerated computing – pushes GPU processing to within a few microseconds of measurement; one paper reports a 3.96 microsecond maximum round trip over commodity Ethernet^[5]. Impressive. Possibly still too slow when the machine's cycle time is a single microsecond and the decoding is non-trivial. Where GPUs may lag, specialized hardware made up of CPUs or FPGAs might step in. The jury's still out on this one.

Practitioners are knocking their heads together to hash it out. ***"It's still quite unclear to us what the use case would be for a very low-latency, tightly integrated quantum system with an HPC system,"*** apart from error correction, admits one quantum-HPC researcher at a major supercomputing center. A quantum director elsewhere frames the opportunity as two hungers at once: using AI to discover new quantum use cases, and using AI to control the machine (calibration, error correction, the fussy work of keeping quantum information intact). What is missing, that director says bluntly, is connective tissue:

"WHAT WE NEED ARE BETTER INTERFACES BETWEEN THE QC AND THE HPC AND AI SYSTEMS. A COMMON VOCABULARY, THAT'S MISSING RIGHT NOW"



Quantum director at a HPC Center.

Experiments and simulations cost little compared to a locked-in architecture built on wrong Quantum-HPC system design assumptions.

So the recommendation writes itself, and it is not a commercial off-the-shelf purchase order. Before committing to a quantum architecture, build the testbed, and better yet: its digital twin. Lay the interfaces and pipelines that feed QPU data to AI and HPC. Then run the experiments and may the best hypothesis wins.

Evaluate before you commit. Letting loose a carrier pigeon may be a joke. Landing a quantum testbed is not. ■

[5] S. A. Caldwell et al., "Platform Architecture for Tight Coupling of High-Performance Computing with Quantum Processors," Nov. 03, 2025, doi: 10.48550/arXiv.2510.25213.

2. UNIFY the software stack: interoperability is the way through

OPEN, MODULAR ECOSYSTEMS ARE THE WAY, NOT FRAGMENTATION AND DUPLICATION.

The RFP responses land on the director's desk, and a fresh headache lands with them. A superconducting machine from one vendor. A neutral-atom system from another. And a trapped-ion system for good measure: sure, why not? Each wrapped in its own software stack, its own scheduler hooks, its own vocabulary, and the center's engineers now have to figure out how to interface all three within the same supercomputer without loss of generality. They rebuild the same job-submission logic three times. They rewrite the same resource descriptions three times. The machines compute in seconds. The plumbing eats months.

This scene is repeating itself across the world's leading computing centers, and it exposes a problem no benchmark has settled. Quantum computing has not yet had its "CMOS transistor" moment, the instant one qubit modality pulls ahead and the rest fall away. That moment may sit years off. Until it arrives, HPC centers have to offer their users a spread of modalities (superconducting, ions, atoms, and more) each carrying distinct strengths, weaknesses, and quirks, and tailor each integration by hand.

Here is what must be done. HPC centers, industry associations, and government programs should join forces now and adopt standards flexible enough to survive a sudden surge, the day a specific modality turns popular and demand for quantum resources outruns supply. Move early, and HPC holds the pen. Wait, and the leverage slips away; once a vendor gains the upper hand, imposing adequate open standards gets far, far harder.

Standardization opportunities can be found in problems independent from qubit modality: declaring a quantum resource in a HPC scheduler, submitting a workload to a quantum computer, or describing a hybrid program.

The fragmentation has run amok. A recent survey by openQSE (a coalition pulling in Oak Ridge, Lawrence Berkeley, Argonne, RIKEN, and vendors from IBM to HPE to Alice & Bob, among others) dissected nine production quantum-HPC stacks and found them walled off from one another. **"Current stacks are isolated, often proprietary, full-stack solutions lacking common interfaces across runtime, resource management, orchestration, and execution layers,"** the group wrote^[6]. Centers duplicate integration logic. Comparisons stall. Portability erodes.

The community trying to manage this mayhem is hit with three problems at once. End users struggle to move a workload from one modality to the next. Administrators multiply custom deployments, one per vendor. Vendors burn engineering hours re-solving problems their rivals already solved.

These pains have some calling to standardize now!

But the counter-pressure pushes just as hard: don't clamp down too early. QPUs behave nothing like racks and cores. Qubits are scarce and tightly coupled; you cannot slice them into fractional allocations, cannot copy the quantum state, cannot checkpoint a job and resume it later. Freeze the design too soon, and you strangle the exploration still underway. As Palsberg and colleagues warn, **"Early consolidation around a single architecture, code family, programming model, or tool-chain risks narrowing the design space before the most important trade-offs are well understood."**^[7]

HPC has run this experiment before. In the early 1990s, High Performance Fortran (HPF) promised to lift programmers above the grind of shuttling data between processors. The abstraction read beautifully but buried the exact knobs users needed. Debugging turned brutal. Performance-hungry scientists dropped back to MPI to claw back control, and HPF never won a lasting community. Abstract away a modality's specifics today, while quantum machines merely crawl, and you mask the very optimizations that make them worth running.

[6] A. Shehata et al., "Quantum-HPC Software Stacks and the openQSE Reference Architecture: A Survey," Jun. 15, 2026, doi: 10.48550/arXiv.2604.20912.

[7] J. Palsberg et al., "Computer Science Challenges in Quantum Computing: Early Fault-Tolerance and Beyond," Jan. 28, 2026, doi: 10.48550/arXiv.2601.20247.



➤ So where does standardization pay off right now? Anywhere that doesn't depend on which qubit modality sits underneath. Declaring a quantum resource to a scheduler. Submitting a job. Accounting for the run. openQSE's reference architecture cuts those boundaries cleanly separating the classical control node, the quantum API, and the quantum resource so each layer evolves behind a stable interface. Two efforts already run in the open: QDMI (Quantum Device Management Interface)^[8], which handles device discovery, and QRMI (Quantum Resource Management Interface)^[9], which exposes QPUs to workload managers like Slurm and Flux. **"The goal is explicitly not to replace existing SDKs, vendor stacks, or open-source efforts,"** the coalition notes, but to draw vendor-neutral lines so the pieces interoperate while sites keep their deployment flexibility.^[6]

A shared standard for integrating quantum resources into common resource managers will emerge from whoever builds it first: HPC centers, vendors, publicly funded research organizations, and academia have every reason to build it together.

The work is already underway. A consortium of national labs and quantum vendors led by IBM recently validated QRMI integration with a broad range of workload managers, including Slurm, PBS, LSF, Grid Engine, Kubernetes, and the Flux Framework, encompassing traditional batch schedulers, a cloud-native orchestration platform, and a graph-based scheduler.^[10] Multiply that handoff across modalities, across sites, across a decade, and the seams start to matter more than the substrate.

One lead quantum-HPC software developer named the reckoning ahead:

"I'D HOPE THAT AT SOME POINT WE GET TOWARDS SOME KIND OF STANDARDIZATION THAT WE CAN ALL BUILD ON. RIGHT NOW, EVERY SINGLE HARDWARE VENDOR HAS THEIR OWN STACK. BUT THIS IS A HILL A LOT OF HARDWARE SUPPLIERS ARE GOING TO END UP DYING ON, BECAUSE THEY DON'T HAVE THE TALENT OR BUDGET."



Quantum-HPC lead developer at a software institute.

HPC centers that grapple with these challenges can tie the ecosystem together and shepherd it to the summit, so many more quantum modalities can be kept in scope for the benefit of scientific end users. ■



[8] Munich Quantum Software Stack, "QDMI — Quantum Device Management Interface, v1.3.2," documentation, accessed Aug. 2026. [Online]. Available: <https://munich-quantum-software-stack.github.io/QDMI/v1.3.2/>

[9] Qiskit Community, "QRMI — Quantum Resource Management Interface," documentation, accessed Aug. 2026. [Online]. Available: <https://qiskit-community.github.io/qirmi/>

[10] T. Badts et al., "Examining QRMI as a Unified Interface for Quantum-HPC Integration," Aug. 08, 2026, doi: 10.48550/arXiv.2607.19591.

3. CO-DESIGN the HPC-Quantum-AI workflows: users care about applications, not qubits

FTQC WON'T BE A DROP-IN ACCELERATOR.

A lead scientist walks into the director's office and asks: what will this new machine do for my team's simulations? The director has fielded that question for years; about vector processors, about GPUs, about accelerators of every stripe. This time the honest answer is: nothing yet. And it can't by itself. A quantum processor awaits on the roadmap, vendors keep arriving with reference architectures and pointed sales decks, and the scientist across the desk does not care about any of it. As one quantum director at an HPC center put it: **"The vast majority of HPC users do not care about the intellectual concept of quantum computing. They care about their application."**

That gap, between a quantum processor and a working application, is where the next few years get won

Full-stack co-design remains largely untapped. Yet it has already delivered a four-orders-of-magnitude reduction in run time, at lower physical qubit count^[11].

or lost. The people running the world's supercomputing centers face a decision they cannot out-source: start co-designing real domain workflows now, before useful QPUs are broadly available, and do it with cross-functional teams that seat domain scientists, HPC and AI experts, and quantum vendors around the same problem. Wait for the hardware, and the workflow won't be ready when the hardware finally lands. ►►

Bringing HPC & AI domain experts, quantum software developers, and quantum hardware specialists together to co-design workflows

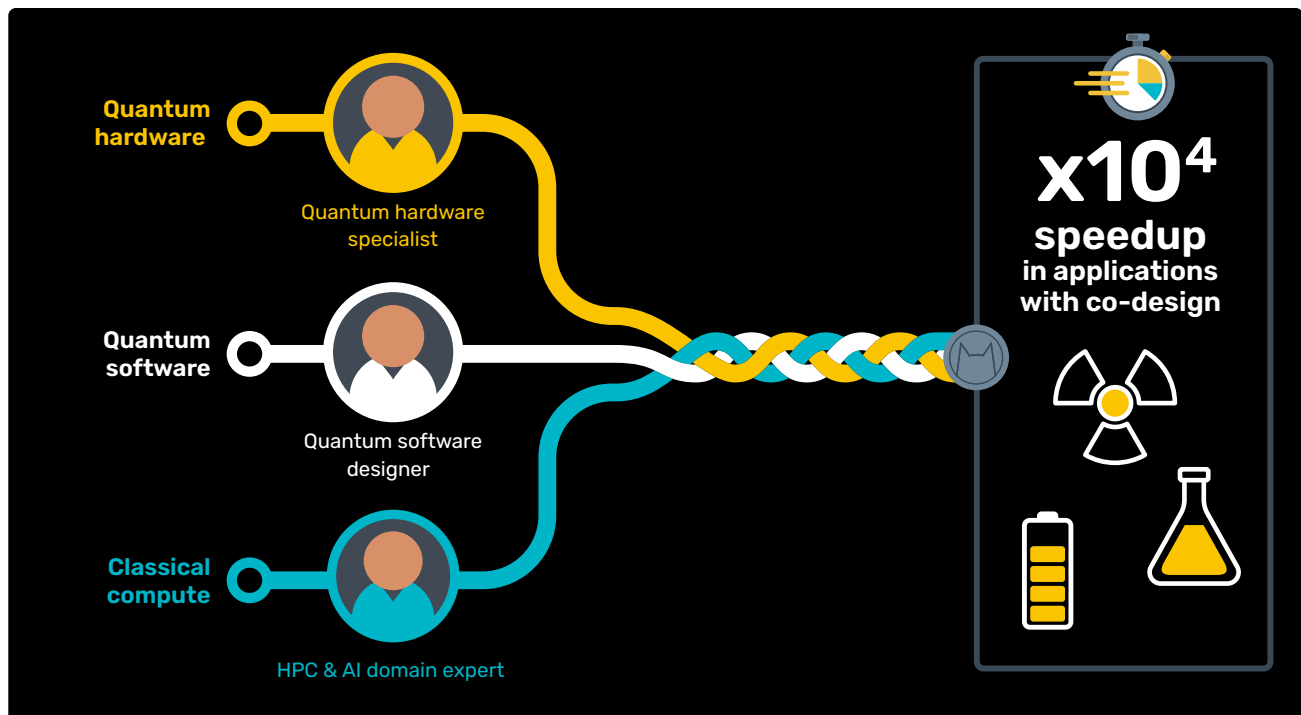
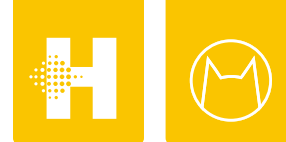


Figure 4 - Bringing a set of different expertises around the same table can decrease the run time of hybrid classical-quantum computing applications by orders of magnitude. 10⁴ speedup comes from the Gratsea and Otten paper^[11].

[11] K. Gratsea and M. Otten, "Achieving Utility-Scale Applications through Full Stack Co-Design of Fault Tolerant Quantum Computers," Nov. 19, 2025, doi: 10.48550/arXiv.2510.26547.



➤ Consider what actually stands in the way. A GPU earns its place the familiar way: you port a kernel, you clock the speedup, you move on. A QPU's nature prohibits that pact. It runs an entirely different programming model, and someone with deep domain knowledge has to decide which slice of a problem could even benefit from quantum acceleration in the first place. Then comes the labor: encoding the problem, preparing the quantum state, wrapping the whole thing in classical pre- and post-processing, orchestrating every hand-off. That is where the effort piles up today, and where it's held up. ***"A lot of co-design effort is needed to run applications on current hardware," a quantum and HPC engineer admits, and it "doesn't seem quite ready yet for general purpose use for a domain scientist."***

The field is blunt about what "ready" means. A workflow counts for nothing until someone validates it, verifies it, and benchmarks it against the best classical methods available (the yardstick HPC users reach for already). And it has to serve more than one kind of user at once. The domain scien-

tist wants the quantum machinery pushed out of sight: hide how the qubits (mis)behave, hide how errors get corrected. The workflow developer and the quantum expert want the reverse. They need to reach down through the stack, all the way to the error-correction layer, and turn the knobs.

The payoff for designing workflows holistically, rather than modularly, is starting to surface in the numbers. A November 2025 study by Gratsea and Otten folded optimizations from three layers (quantum hardware, fault-tolerant architecture, and algorithmic subroutines) into a single co-design framework and cut runtime by nearly four orders of magnitude at lower physical qubit counts^[1]. Their conclusion cuts deeper than their savings. No single innovation, pursued alone, would have moved the needle. The gains were eked from the seams between disciplines.

Funding agencies have noticed. The U.S. Department of Energy's ARPA-E program, through its Quantum Computing for Computational Chemistry effort, ➤

Full stack co-design accelerates the timeline to usefulness

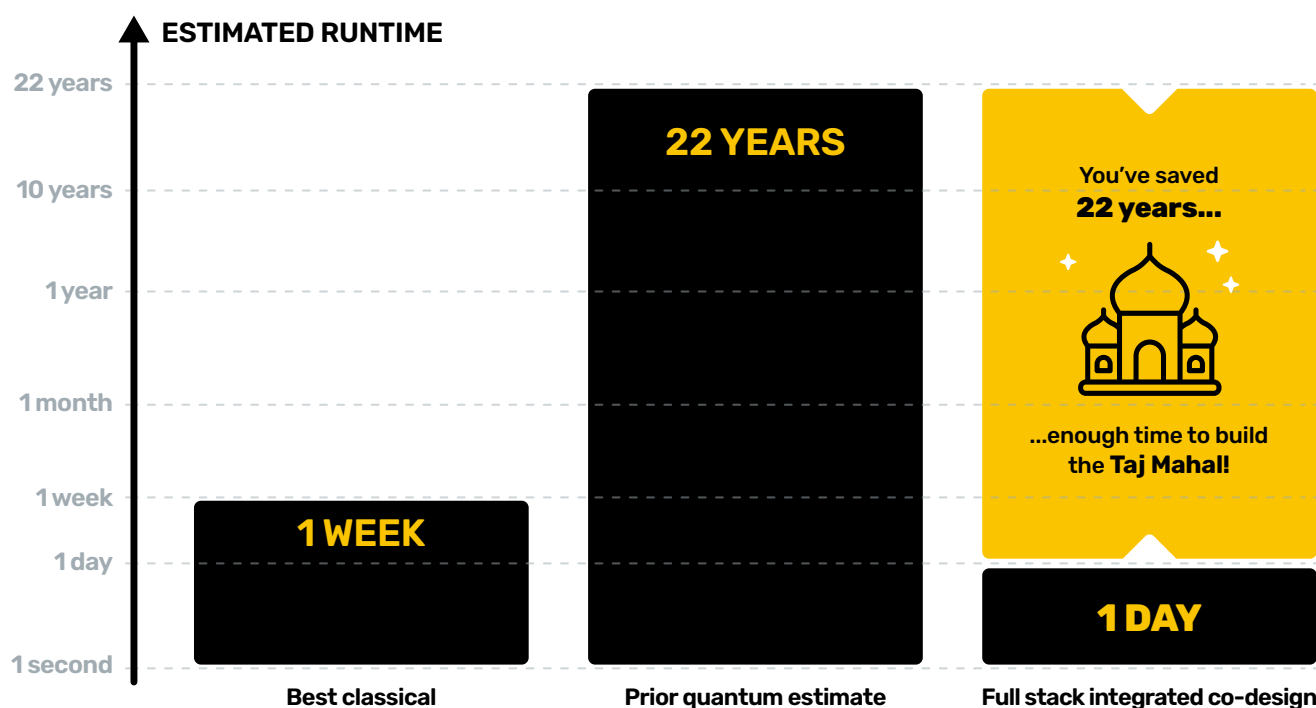


Figure 5 - Estimated runtime for the same quantum chemistry problem: full-stack quantum-classical co-design turns a 22-year quantum runtime into 1 day, now with the potential to outpace the best classical method in the near-term.^[1]

➤ now treats quantum advantage as a joint hardware-and-software outcome rather than just a hardware milestone. In one project, Alice & Bob works alongside Los Alamos, the University of Michigan, and GE Vernova on a chemistry workflow: classical methods compute the environmental parameters that condition a quantum simulation of a molecule's strongly correlated electronic structure, and the quantum state preparation gets designed around the shape of that classical output, not bolted on afterward^[12]. The target reads like a dare: a 10,000-fold reduction in time to solution against the best classical simulation. Whether it unfolds depends entirely on the two stages being built as one.

Scientific workflows co-designed today, by cross-functional teams around one real problem, will be the ones ready to capture the full potential of useful QPUs when they arrive.

AI threads into these workflows too, no longer buried in decoding and calibration but surfacing at the application level, in the pre- and post-processing stages where the latency demand loosens its grip. How to wire it efficiently remains an open question. The interfaces don't exist yet. Neither do the workflows. Cross-disciplinary teams have to build them.

"A LOT OF CO-DESIGN EFFORT IS NEEDED TO RUN APPLICATIONS ON CURRENT HARDWARE."



A quantum and HPC engineer.

So the work in front of HPC leaders is not to wait, and not to abstract everything away before anyone understands the trade-offs. It is to seat the right people from across disciplines around one real problem and start today. The domain scientist's question hasn't changed in decades. Now as then, the answer has to be built before the machine arrives. ■

[12] Alice & Bob, "Alice & Bob Secures \$3.9M ARPA-E Award," newsroom, accessed Aug. 2026. [Online]. Available: <https://alice-bob.com/newsroom/alice-bob-secures-3-9m-arpa-e-award-to-use-quantum-computing-to-design-rare-earth-free-magnets/>



EXPERTISE IS BUILT IN-HOUSE: The case for on-premise deployment

REACH OUT AND BUILD IN-HOUSE.

Step back, and a pattern emerges across every front. The barrier to useful quantum computing in HPC no longer sits at the qubits. It lurks in the linkages, the hybrid architecture, the software layers, the workflows nobody has co-designed. None of that ships in a crate with the mature machine. It accrues over years, forged by the centers wrestling with the open questions instead of waiting them out.

So every director comes to the same fork: build readiness in the cloud, or on-prem?

Cloud beckons first. It lets a center stand up software, gather users, and shop modalities before a dollar is committed to hardware. But you're left blind to what's behind the API. Reach only as deep as the vendor allows, and someone else still runs the machine; the lessons of operating it stay with them. One senior researcher draws the line: **"If you have interacted with quantum computers**

only through the cloud, you have no idea how a quantum computer stack looks. All you see is Python, and that's it. Having root access on the control node of a quantum computer is a whole different experience."

History rhymes. When GPUs landed, the centers that racked them early wrote the programming models and benchmarks everyone else inherited. The standards took shape where the machines hummed.

An on-prem QPU is that testbed, where assumptions stop being argued and start being assessed. As one quantum lead puts it: **"Developing something, getting your hands dirty, it really helps in understanding what the problem is. It's part of getting a seat at the adults' table."**

The first fault-tolerant machines will go to the centers already reaching. Readiness is built now. ■

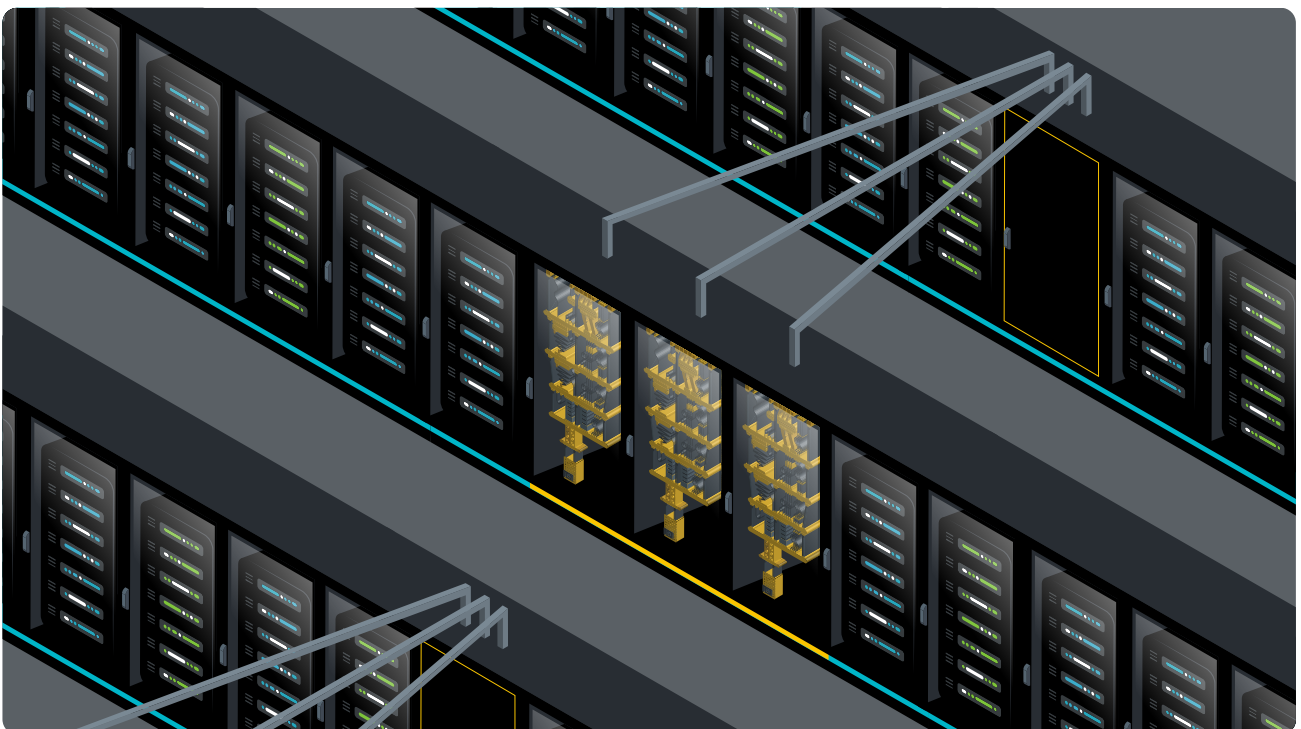


Figure 6 - On-premise deployment builds the expertise that underpins quantum leadership.

References

- [1] S. Seelam et al., "Reference Architecture of a Quantum-Centric Supercomputer," Mar. 12, 2026, doi: 10.48550/arXiv.2603.10970.
- [2] C. Gidney, "How to factor 2048 bit RSA integers with less than a million noisy qubits," May 21, 2025, doi: 10.48550/arXiv.2505.15917.
- [3] D. Camps, E. Rrapaj, K. Klymko, B. Austin, and N. J. Wright, "Evaluation of the Classical Hardware Requirements for Large-Scale Quantum Computations," in ISC High Performance 2024 Research Paper Proceedings (39th International Conference), Hamburg, Germany: IEEE, May 2024, pp. 1–12. doi: 10.23919/ISC.2024.10528937.
- [4] K. Kissell, "Decoupling AI for Quantum Control and Calibration", Jun. 2026, Available: <https://alice-bob.com/blog/decoupling-ai-for-quantum-control-and-calibration/>
- [5] S. A. Caldwell et al., "Platform Architecture for Tight Coupling of High-Performance Computing with Quantum Processors," Nov. 03, 2025, doi: 10.48550/arXiv.2510.25213.
- [6] A. Shehata et al., "Quantum-HPC Software Stacks and the openQSE Reference Architecture: A Survey," Jun. 15, 2026, doi: 10.48550/arXiv.2604.20912.
- [7] J. Palsberg et al., "Computer Science Challenges in Quantum Computing: Early Fault-Tolerance and Beyond," Jan. 28, 2026, doi: 10.48550/arXiv.2601.20247.
- [8] Munich Quantum Software Stack, "QDMI — Quantum Device Management Interface, v1.3.2," documentation, accessed Aug. 2026. [Online]. Available: <https://munich-quantum-software-stack.github.io/QDMI/v1.3.2/>
- [9] Qiskit Community, "QRMI — Quantum Resource Management Interface," documentation, accessed Aug. 2026. [Online]. Available: <https://qiskit-community.github.io/qirmi/>
- [10] T. Badts et al., "Examining QRMI as a Unified Interface for Quantum-HPC Integration," Aug. 08, 2026, doi: 10.48550/arXiv.2607.19591.
- [11] K. Gratsea and M. Otten, "Achieving Utility-Scale Applications through Full Stack Co-Design of Fault Tolerant Quantum Computers," Nov. 19, 2025, doi: 10.48550/arXiv.2510.26547.
- [12] Alice & Bob, "Alice & Bob Secures \$3.9M ARPA-E Award," newsroom, accessed Aug. 2026. [Online]. Available: <https://alice-bob.com/newsroom/alice-bob-secures-3-9m-arpa-e-award-to-use-quantum-computing-to-design-rare-earth-free-magnets/>
- [13] M. Mohseni, A. Scherer, K. G. Johnson, et al., "How to Build a Quantum Supercomputer: Scaling from Hundreds to Millions of Qubits," Nov. 2024, doi: 10.48550/arXiv.2411.10406.
- [14] Open Compute Project Foundation, "Integrating Quantum Processing Units into Data Center Infrastructure," white paper, accessed Aug. 2026. [Online]. Available: <https://www.opencompute.org/documents/integrating-qpus-into-dc-infrastructure-pdf>
- [15] P. Hopf, S. Stern, R. Wille, and L. Burgholzer, "Standardizing Access to Heterogeneous Quantum Backends: A Case Study on Cloud Service Integration with QDMI," Mar. 2026, doi: 10.48550/arXiv.2603.05138.
- [16] L. Burgholzer, M. Walter, P. Hopf, et al., "Practical HPCQC Integration with QDMI: A Real-Hardware Case Study with IQM Systems," Apr. 2026, doi: 10.48550/arXiv.2604.19869.
- [17] P. Rao et al., "Performance Model for Hybrid Quantum-Classical Workflows," Jul. 16, 2026, doi: 10.48550/arXiv.2607.15426.

AUTHORS

• Hyperion Research •
Bob Sorensen

• Alice & Bob •

Mamdouh Abbara, Niccolo Coppola, Eliott Doutriaux, Charbel Eid, Imanol Hernandez, Paul Kassebaum, Victor Kerros, Kevin Kissell, Juliette Peyronnet, Laurent Prost, Wes Roberts, Elsa Venancio, Rémi de la Vieuvville.

ACKNOWLEDGEMENTS

In the course of preparing this report, we conducted interviews with experts from Argonne National Laboratory, Lawrence Berkeley National Laboratory / NERSC, Oak Ridge National Laboratory, AIST / G-QuAT, LINKS Foundation, Pawsey Supercomputing Centre, Munich Quantum Valley / Munich Quantum Software Company, CEA, OCP, RIKEN, Marqov, SURF, bp, Dell, and GE Vernova.

We warmly thank them for the insightful discussions on the present and future of HPC-Quantum-AI integration, which greatly enriched our understanding of the field. The views, analyses, and conclusions presented in this report are solely those of the authors and do not necessarily reflect the opinions or positions of the interviewees or their respective institutions.

ABOUT HYPERION RESEARCH

Hyperion Research provides data-driven research, analysis and recommendations for technologies, applications, and markets in high performance computing and emerging technology areas to help organizations worldwide make effective decisions and seize growth opportunities. Research includes market sizing and forecasting, share tracking, segmentation, technology, and related trend analysis, and both user and vendor analysis for multi-user technical server technology used for HPC and HPDA (high performance data analysis). Hyperion Research provides thought leadership and practical guidance for users, vendors, and other members of the HPC community by focusing on key market and technology trends across government, industry, commerce, and academia.

More information at: www.HyperionResearch.com

ABOUT ALICE & BOB

Alice & Bob is a quantum computing company based in Paris and Boston whose goal is to create the first universal, fault-tolerant quantum computer. Founded in 2020, Alice & Bob has already raised €180 million in funding, hired over 250 employees and demonstrated experimental results surpassing those of technology giants such as Google or IBM. Alice & Bob specializes in cat qubits, a pioneering technology developed by the company's founders and later adopted by Amazon. Demonstrating the power of its cat architecture, Alice & Bob recently showed that it could reduce the hardware requirements for building a useful large-scale quantum computer by up to 200 times compared with competing approaches.

Follow Alice & Bob on LinkedIn, X or YouTube, visit their website www.alice-bob.com.

