

An extensible quantum network architecture

How quantum computers are joined together with light, and what it takes to make that work at commercial scale.

Date: September 2026

Control#: MRA-QNRA-091726.1

Authors: Dr. Shobhit Gupta, memQ Inc; Dr. Sean Sullivan, memQ Inc.; Dr. Manish Singh, memQ Inc.

Contents

Contents	2
1. Executive Summary	2
2. Why scale comes from connecting machines	2
3. What a qubit is, and why you cannot just send it over a classical network	3
4. Photon-mediated entanglement	4
5. The QNIC: Edges of the network	5
6. The Entanglement Hub: Center of the network	7
7. Precision control	8
8. The distributed quantum compiler	8
9. Putting it together	9
Proof points	11
Where to Find Additional Information	11
Acknowledgments	12
Copyright	12
Trademark Information	12

1. Executive Summary

The path to a commercially useful quantum computer runs through the network, not through a bigger chip. Every qubit technology in production today – superconducting, trapped ion, neutral atom, electron spin – is running into the same wall: the control wiring, cooling budget, and crosstalk of a single processor all grow faster than the useful qubit count does. Classical computing met this wall in the 1990s and went around it by connecting machines together. Quantum computing is arriving at the same answer.

The thing that connects two quantum processors is light. A photon carries quantum information between them, and when photons from two machines are made indistinguishable and measured together, the two machines end up entangled – linked in a way that lets them run one computation across both. This is called photon-mediated entanglement, and it has been demonstrated between processors in separate buildings.

Making it work at a commercial scale takes four things, and this paper explains each one. A **quantum network interface controller (QNIC)** at each processor, which gets quantum information out of the qubits and onto a photon the network can carry. An **Entanglement Hub** where photons from different processors meet, are held if they arrive early, and are measured together. **Precision quantum control**, because the entire operation happens inside a time window a few tens of nanoseconds wide. And a **distributed compiler**, which determines how to split a problem across machines that are connected this way.

memQ builds the parts of that chain that are both hard and specific to a given qubit technology, in commercial photonic foundries, so that they can be supplied to any quantum computer maker rather than locked inside one company's stack.

2. Why scale comes from connecting machines

Building a large-scale, useful quantum computer is a monumental engineering task. Many applications for useful quantum algorithms anticipate the need for large numbers (on the order of 100,000s to 1M) of physical qubits to be available. This is due to error correction – because of the unique nature of quantum information, as we will show later, multiple physical qubits need to be used to produce one logical ‘error corrected’ qubit that can be used in calculations.

And while ultimately, each qubit type has its own key strengths and challenges, the qubits themselves are not necessarily the biggest challenge in scaling. It is everything attached to them. Each qubit needs control lines, optical or microwave signals, and isolation from its neighbors. Doubling the qubit count roughly doubles the wiring, the cooling load, and the number of ways adjacent qubits can disturb each other. Past a certain size, adding qubits to one processor costs more performance than it buys.

Classical computing hit this wall and found a way around it. Today’s systems no longer attempt to build a supercomputer as one enormous processor; they build racks of modest machines and spend the engineering effort to connect them together. This interconnect technology is crucial to modern computing and is now a major driver for advancements in expanding AI training capabilities and high performance computing.

The same logic applies here, with one addition. Quantum technologies are not all interchangeable: trapped ions hold quantum states for a long time, neutral atoms scale well in number, and superconducting circuits switch fast. A network is what lets a workload use more than one of them. A useful test for whether an architecture is genuinely modular: *does adding a new component require redesigning the network and its protocols?* If it does, it is not modular, and it will not scale.

Fortunately, this is no longer a thought experiment for quantum systems. In 2025 a group at Oxford ran a quantum algorithm across two trapped-ion processors in separate systems, linked only by an optical connection. This was the first demonstration of a distributed quantum computation over a photonic link.

Citations for this section. Ang et al., *ARQUIN: Architectures for Multinode Superconducting Quantum Computers*, *ACM Trans. Quantum Comput.* **5**, 1–59 (2024), [10.1145/3674151](#). Main et al., *Distributed quantum computing across an optical network link*, *Nature* **638**, 383–388 (2025), [10.1038/s41586-024-08404-x](#).

3. What a qubit is, and why you cannot just send it over a classical network

A classical bit is either 0 or 1. A qubit can hold a weighted combination of both at once, and a group of qubits can hold a combination of all their possible states together. Running a calculation on that combination, rather than on one value at a time, is where quantum computing’s advantage comes from.

Physically, a qubit is any object with two states you can reliably prepare, manipulate, and read: two electronic states of a trapped ion, two hyperfine states of a neutral atom, a single electron spin in a crystal, the two lowest energy levels of a superconducting circuit. While the different qubit modalities look different, each can be coaxed into emitting a photon whose state is tied to the qubit’s own state.

For quantum systems, there are two key constraints:

- **A quantum state cannot be copied.** (No-cloning Theorem) This is a fundamental limitation posed by quantum mechanisms.
- **Reading a quantum state destroys it.** (Observer Effect, Wave Function Collapse) A classical network handles loss by amplifying and repeating the signal, which means reading it and sending a fresh copy. Do that to a qubit and the quantum information is lost or corrupted.

So a quantum network cannot be built the way a classical one is. The state has to travel via a single photon, once, and if that photon is absorbed or scattered on the way, the attempt is simply lost. Every design decision in this architecture — the choice of wavelength, the use of integrated optics, where the detectors sit — traces back to one goal: lose as few photons as possible, and know immediately when you did.

The network's primary goal is therefore transmitting qubit states to generate **entanglement**: two qubits, one in each processor, placed in a shared state such that operations on one affect the other. Once two machines share entanglement, they can use it to run gates between them. Entanglement is the commodity; the rest of this paper is about manufacturing and delivering it.

Citations for this section. Wootters & Zurek, *A single quantum cannot be cloned*, *Nature* **299**, 802–803 (1982), [10.1038/299802a0](https://doi.org/10.1038/299802a0). Main et al., *Distributed quantum computing across an optical network link*, *Nature* **638**, 383–388 (2025), [10.1038/s41586-024-08404-x](https://doi.org/10.1038/s41586-024-08404-x).

4. Photon-mediated entanglement

Since a qubit cannot be copied or transmitted reliably, the network does something indirect instead. Each processor keeps its matter-based qubit and sends out a photon that is entangled with it. The two photons meet in the middle, and a measurement performed there transfers the link to the two qubits that stayed in their respective QPUs.

In five steps:

1. **Emit.** A qubit in processor A is excited so that it emits a single photon whose state is entangled with the qubit's state. Processor B does the same. Neither photon carries a copy of anything; each is one half of a local entangled pair.
2. **Travel.** Both photons are sent down optical fiber to a common midpoint.
3. **Erase.** At the midpoint the two photons are made indistinguishable. This includes the same wavelength, same shape in time, same polarization, same spatial mode. The photons are combined on a beam splitter. Any property that would reveal which photon came from which machine has to be removed – the “which-way” information is erased.
4. **Measure.** Single-photon detectors positioned after the beamsplitter register the arrival. Certain click patterns are only possible if the “which-way” information is truly gone.
5. **Herald.** That click pattern is announced back to A and B over a classical channel. The two origin qubits that stayed in their base QPUs are now entangled, and the machines can use the link.

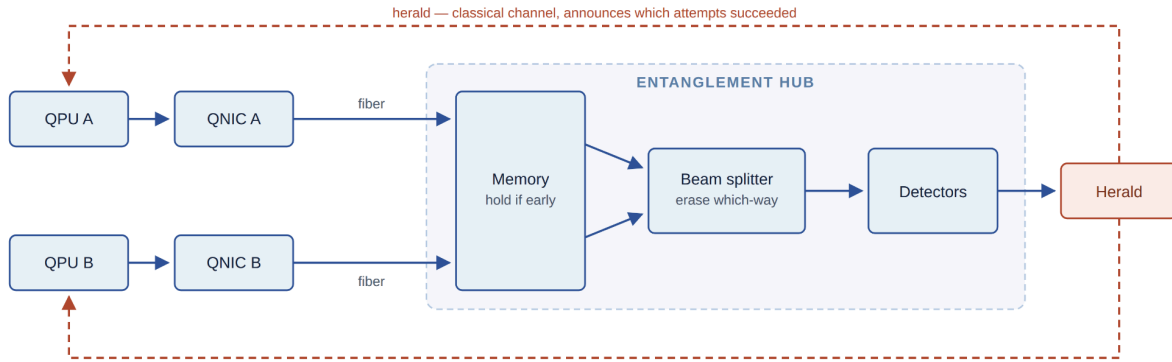


Fig. 1: Schematic showing for photon mediated entanglement between processors, QPUs A and B. Photons emitted from each travel to an entanglement hub where their which-way information is erased and their coincidence on a beamsplitter is measured. The resulting detection events are heralded and sent back to the QPUs for further operations.

The entanglement connection here is **heralded**, meaning its success isn't guaranteed, but successful attempts are noted. Many attempts fail, usually because a photon was either not emitted by the QPU, lost in the fiber or missed by a detector. The particular pattern of clicks on the detectors tells you which entanglement attempt between processors succeeded. Once success is heralded, that information is relayed back to the QPUs to do further operations on the qubits. The practical consequence is that entanglement rate – the number of successful links per second – is a key performance number for a quantum network, similar to the way bandwidth is for a classical one.

Two photons interfere only if a detector cannot distinguish them, i.e. the “which-way” information has been erased. Photons from different qubit technologies naturally differ in wavelength and in how long they last: for instance a rubidium atom emits near 780 nm with a bandwidth of roughly 6 MHz, corresponding to a photon about 26 ns long. The QNIC needs to ensure that the wavelength, shape, polarization of photons coming from the different QPUs are identical.

Photon mediated entanglement has been demonstrated well outside the lab. Two single rubidium atoms have been entangled across 33 km of telecom fiber, and two solid-state memory nodes have been entangled over 35 km of fiber deployed around the Boston metropolitan area.

Citations for this section. *van Leent et al., Entangling single atoms over 33 km telecom fibre, Nature* **607**, 69–73 (2022), [10.1038/s41586-022-04764-4](https://doi.org/10.1038/s41586-022-04764-4). *Knaut et al., Entanglement of nanophotonic quantum memory nodes in a telecom network, Nature* **629**, 573–578 (2024), [10.1038/s41586-024-07252-z](https://doi.org/10.1038/s41586-024-07252-z), *Pompi et al., Realization of a multinode quantum network of remote solid-state qubits, Science* **372**, 259-264 (2021), [10.1126/science.abg1919](https://doi.org/10.1126/science.abg1919)

5. The QNIC: Edges of the network

A classical computer connects to a network through a network interface card. It takes data the machine holds in its own internal format and puts it on the wire in the format the network expects. A quantum

processor needs the same thing, and the job is considerably harder because the payload is a single photon that cannot be regenerated if it is mishandled.

The entity to perform this function is called the **quantum network interface controller**, or QNIC (a term first coined in 2014 and codified in a number of software frameworks such as QuISP). A QNIC should accomplish four functions:

Job	What it does	Engineering Challenges
Extraction	Drives a qubit to emit a photon entangled with it, and collects that photon into a fiber	Emission happens in every direction; whatever is not collected is lost, and lost photons are the dominant error
Transduction	If needed, shifts the photon's wavelength to the network standard without disturbing its quantum state	The conversion has to be efficient and add essentially no noise, or it adds errors
Transcoding	If needed, converts the photon into the network's standard encoding, and shapes it in time so it can interfere with a photon from a different machine	Two photons must become indistinguishable while each stays entangled with its own qubit
Switching	Routes the photon onto the right network path, at the right instant	The decision window is nanoseconds wide

Extraction is an optics problem before it is a quantum problem. A qubit that emits into free space sends light everywhere; unless it is surrounded by something that channels the emission, most of it never reaches a fiber. The answer is to build optical structure around the qubit: a cavity, a high numerical aperture lens, or for solid-state qubits a photonic chip that the qubit sits inside. Collection efficiency set here caps everything downstream, which is why it is the first number to consider in your quantum networking architecture.

Transduction exists because qubits emit at the wavelengths physics gives them, and those are almost never the wavelengths fiber likes. Rubidium emits near 780 nm, where fiber loss is roughly an order of magnitude worse than in the telecom C-band. The C-band near 1.53–1.57 μm has the lowest loss available in optical fiber (0.2 dB/km), meaning about 4.5% of photons are lost per kilometer. This wavelength range also has thirty years of commercial component development behind it: tunable lasers, low-loss filters, and foundry integrated photonics. Converting a photon into that band, and back, is the single largest reach-extension available. Single atoms have been entangled over 33 km of telecom fiber, demonstrating that the conversion can be done without destroying the quantum state.

Transcoding is the step that makes two machines compatible. The network fixes one encoding — memQ's general choice is time-bin, which survives fiber well and works naturally with precision timing — and the QNIC converts whatever the local qubit produces into it. It also shapes the photon in time, so that a photon from a fast emitter and one from a slow emitter can still be made to interfere. This is where the multimodal bandwidth-matching problem is solved or not solved, and it is the most common technical question memQ is asked.

Switching is the part that looks most like classical networking and behaves least like it. There is no buffer to hold a photon while a route is computed. The path has to be set before the photon exists, which means the switching decision belongs to the control layer described in Section 7, not to the network stack in the usual sense.

The QNIC divides cleanly into a part that depends on the qubit technology — extraction and the first stage of transduction — and a part that does not, because it faces the network. memQ's approach is to build the modality-specific part as photonic chiplets fabricated in commercial foundries, which drop into an otherwise common design. The consequence is commercially important: the same network, the same hub, and the same control interface serve a trapped-ion machine and a neutral-atom machine, and a customer is not required to buy their qubits and their network from the same company.

***Citations for this section.** Pettit, Farshi, Sullivan, Véliz-Osorio & Singh, A perspective on the pathway to a scalable quantum internet using rare-earth ions, Appl. Phys. Rev. **10**, 031304 (2023), [10.1063/5.0156874](https://doi.org/10.1063/5.0156874). van Leent et al. (2022), as above, for quantum frequency conversion to telecom in a working link. [CITE — add a high-efficiency cavity-enhanced collection reference for extraction, and a foundry/PDK integrated-photonics reference for the chiplet claim.]*

6. The Entanglement Hub: Center of the network

Processors do not entangle with each other directly. They each send a photon to a shared facility, and that facility does the work. memQ calls it an **Entanglement Hub**, and it has four jobs:

- **Receive.** Photons arrive from several nodes at once, on separate fibers, and have to be routed toward a common measurement point without being lost on the way. Every optical element between the fiber and the detector costs photons, so the hub is built from low-loss integrated optics rather than assembled from discrete components.
- **Hold.** Photons from two machines will not arrive at the same instant. The nodes sit at different distances, they emit at different moments, and emission is probabilistic in the first place. A quantum memory absorbs a photon that arrives early and releases it on command, so that it can meet a partner that arrives later. Without memory, the network only succeeds when two independent low-probability events happen to coincide; with it, the two events only need to happen within the memory's storage time. That is the difference between a demonstration and a service. A memory useful here must store many photons at once, fit in a small enough footprint that many can sit side by side, release on demand rather than after a fixed delay, and accept a wide enough range of photon bandwidths to serve more than one qubit technology.
- **Release together.** The photons are brought onto a beam splitter in the same instant and the same spatial mode, so that no measurement afterwards could reveal which machine each came

from. This is the erasure step from Section 4, and it is where the timing requirement in Section 7 comes from.

- **Measure.** Single-photon detectors register the outcome. The hub then announces the result back to the two nodes as fast as physically possible, because the qubits at each end are holding a fragile state while they wait. memQ's approach sends that announcement as a classical optical pulse down the same dedicated timing fiber, rather than reconstructing the coincidence after the fact from timestamps at each end.

A key reason to centralize the entanglement hubs is cost. The best single-photon detectors and the memories both need cryogenic cooling. In this approach, we consolidate those cryogenic components into a single system to minimize capital cost, floor space, and the number of components that can fail. A network with cryogenics consolidated into shared hubs scales the way data center infrastructure does: nodes stay comparatively simple, and the expensive equipment is amortised across everything connected to it. Crucially, the choice of the detector is the main factor here – superconducting detectors outperform semiconductor ones in terms of efficiency and dark count rate.

The second reason is standardization. If every hub accepts the same range of wavelengths, similar bandwidth, and the same encoding, then adding a node to the network is a matter of connecting a fiber.

***Citations for this section.** Pettit, Farshi, Sullivan, Véliz-Osorio & Singh (2023), as above, for the quantum memory requirements and the rare-earth approach. Knaut et al. (2024), as above, for memory nodes operating over deployed telecom fiber.*

7. Precision control

Everything in Sections 5 and 6 has to happen inside a window narrower than most software can address. For example, a rubidium atom's photon is about 26 nanoseconds long. For two such photons to interfere at a beam splitter, they have to overlap there, which means the two emissions, two routing decisions, and arrivals all have to be orchestrated to a small fraction of that duration, across machines that may be many meters apart.

Event	Timescale
Overlap window at the beam splitter	sub-nanosecond to nanoseconds
Photon duration (e.g., rubidium)	~26 ns
Light through 1 km of fiber	~5 μ s
Qubit coherence time, holding a state while waiting	microseconds to seconds, depending on technology
General-purpose operating system scheduling	~1 ms

The timescales between the top and the bottom of the table span 5-6 orders of magnitude. A conventional server that is attempting to orchestrate a quantum device and then correcting it a

millisecond later may be making decisions based on photons that were measured/destroyed before the message was even received. So control is divided into two layers:

The **slow layer** incorporates ordinary infrastructure software. This includes general scheduling: which job runs, what resources are partitioned, and how the network is configured. Timescales are on the order of up to ~milliseconds.

The **fast layer** covers a short list of operations – firing the control lasers (optical modulators), triggering setting the optical switch, releasing a stored photon to the beam splitter. These actions must be executed deterministically, effectively in real time. The fast layer is driven from a common timing reference distributed over dedicated optical fiber, separate from the data network, so that its precision does not depend on network traffic.

The practical point for anyone planning a deployment: quantum networking cannot be fully bolted onto conventional datacenter orchestration. The slow layer can be, and should be. The fast layer is a distinct piece of infrastructure, and an architecture that does not account for it separately will not function properly.

***Citations for this section.** Delle Donne et al., An operating system for executing applications on quantum network nodes, *Nature* **639**, 321–328 (2025), [10.1038/s41586-025-08704-w](https://doi.org/10.1038/s41586-025-08704-w) – highlights the separation between a general-purpose control layer and a real-time one.*

8. The distributed quantum compiler

The standard way to represent a quantum computing program is to use circuits that show the execution of gate operations on the hardware over time. A user is free to program a monolithic quantum computer in this way; however, such an approach fails for distributed architectures that must account for the latency and noise of links. Something has to work out how to spread that program across several machines connected by the links described above, and that something is the distributed compiler.

Generally speaking, there are two ways to do this.

- **Cut it classically.** A circuit can be cut into pieces that run independently on separate processors, with the results stitched back together by classical post-processing – a technique known as circuit knitting or circuit cutting. No actual entanglement is consumed because no quantum link is needed. However, each cut multiplies the number of times the circuit has to be run. This multiplier grows exponentially with the number of cuts, making it highly unscalable for large circuits.
- **Teleport the gate.** A gate between qubits on two different machines can be executed directly by consuming a shared entangled pair. Here, each interconnect needs to generate one entanglement per gate. This has better scaling prospects than cutting, and is what the Oxford demonstration in Section 2 did. However, this only works if the network is delivering entangled pairs at a useful rate and quality.

So the compiler is making an economic decision, per job: is entanglement currently cheap enough to buy, or is it cheaper to pay in classical repetitions? That question has no fixed answer. It depends on

which machines are free, how much loss is on each link right now, what the current entanglement rate and fidelity are, and what the network topology looks like at that moment. A classical compiler does not need to know the instantaneous physical condition of the network it targets. A distributed quantum compiler does, and that is the defining difference.

The sequence is straightforward to describe. Assess the workload and identify where it can be parallelised. Decide, for each boundary, whether to cut or to teleport. Assign the pieces to specific processors, taking account of what each qubit technology is good at — a subroutine that needs long coherence and one that needs fast gates need not run on the same kind of machine. Execute, with the network scheduler reserving entanglement in advance where teleported gates are planned. Recombine the results.

This is also where heterogeneity stops being a slogan and becomes a capability. A network that carries a standard photon format between different qubit technologies lets the compiler place each part of a problem on the hardware that suits it. A vertically integrated stack cannot offer that, because it only has one kind of qubit to offer.

Citations for this section. *Piveteau & Sutter, Circuit knitting with classical communication, [IEEE Transactions on Information Theory 70, 2025](#). Gupta, et al. Gate Teleportation vs. Circuit Cutting in Distributed Quantum Computing, *Advanced Quantum Technologies* 9, no. 9 (2026). <https://doi.org/10.1002/qute.70424>. Main et al. (2025), as above, for a teleported gate executed between separate processors over an optical link. Delle Donne et al. (2025), as above, for the scheduling layer.*

9. Putting it together

A photon leaves each processor, is prepared by a QNIC into a common format, travels over fiber to a shared hub, waits in memory (if it arrives early), meets its partner on a beam splitter, is measured, and the result is announced back. The two qubits remaining in the QPUs are now entangled. A compiler decides when that link is worth spending, and a precision timing layer makes the whole sequence land inside a window a few tens of nanoseconds wide. The full architecture is shown in Fig. 2.

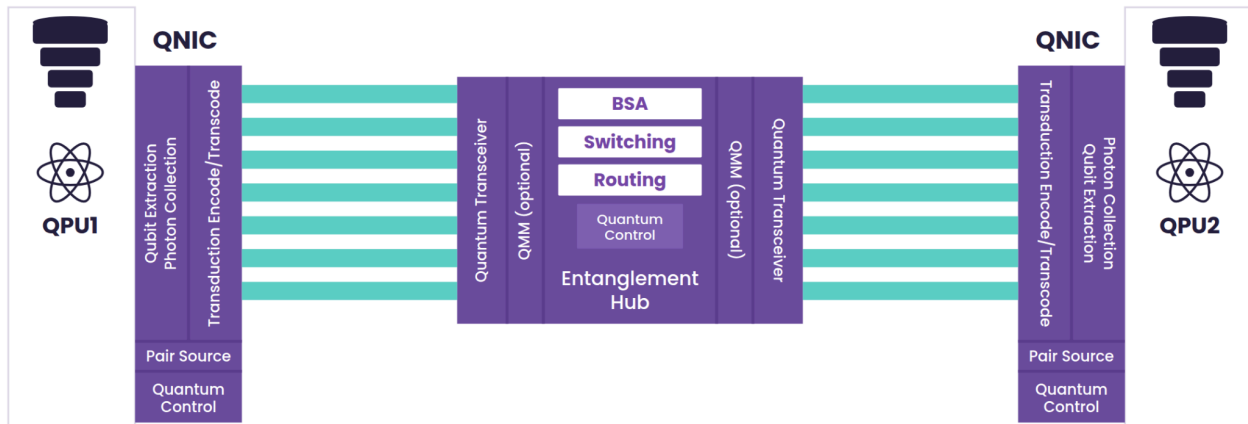


Fig. 2: memQ's extensible quantum network architecture for photon mediated entanglement. The teal lines represent multiplexed fiber connections.

This enables:

- **Capacity by addition.** More computational reach comes from connecting another machine, not from redesigning the one you have.
- **Freedom of supplier.** A standard photon format at the network boundary means the qubit vendor and the network vendor need not be the same company — and a customer is not locked into one technology's ceiling.
- **Shared expensive infrastructure.** Cryogenics, detectors, and memory sit in hubs and are amortised across every node attached to them.
- **One architecture from rack to metro.** The same interfaces that connect two machines in a room extend, over telecom-band fiber, to machines in different buildings — as the 33 km and 35 km results in Section 4 show.

Proto versions of this are being assembled now, most visibly in Illinois around the Q-NEXT hubs at Argonne, UChicago and UIUC and at the Illinois Quantum and Microelectronics Park, with comparable efforts in Montana, New York, and at QuTech in the Netherlands.

Proof points

Claim in this paper	What has been shown	Source
Two distant matter qubits can be entangled using photons	Two single rubidium atoms entangled across 33 km of telecom fiber, using frequency conversion to the telecom band	van Leent et al., Nature 607 , 69–73 (2022), 10.1038/s41586-022-04764-4
It works on real deployed infrastructure	Two nanophotonic memory nodes entangled over 35 km of deployed fiber under metropolitan Boston	Knaut et al., Nature 629 , 573–578 (2024), 10.1038/s41586-024-07252-z
A computation can be run across two processors over an optical link	Distributed quantum algorithm executed across two trapped-ion modules joined by a photonic link	Main et al., Nature 638 , 383–388 (2025), 10.1038/s41586-024-08404-x
A quantum memory can do the hold-and-multiplex job at a hub	Requirements and pathway for rare-earth-ion memories in a scalable quantum network	Pettit, Farshi, Sullivan, Véliz-Osorio & Singh, Appl. Phys. Rev. 10 , 031304 (2023), 10.1063/5.0156874
Control splits into a general-purpose layer and a real-time layer	Operating system executing applications on quantum network nodes, with that separation demonstrated	Delle Donne et al., Nature 639 , 321–328 (2025), 10.1038/s41586-025-08704-w
Multinode architectures have been analysed at system level	Architecture study for multinode superconducting quantum computers	Ang et al., ACM Trans. Quantum Comput. 5 , 1–59 (2024), 10.1145/3674151
Splitting a circuit classically is a plausible option with a known cost	Circuit knitting with classical communication, and its sampling overhead	Piveteau & Sutter, arXiv:2205.00016 ; Gupta et al. AQT 9 , 9 (2026), 10.1002/qute.70424

Where to Find Additional Information

To learn more about the information described in this document, review the following resources:

- R. M. Pettit, F. H. Farshi, S. E. Sullivan, Á. Véliz-Osorio, and M. K. Singh, “A perspective on the pathway to a scalable quantum internet using rare-earth ions,” Applied Physics Reviews, vol. 10, no. 3, 031304, Sep. 2023, doi: 10.1063/5.0156874.

- W. K. Wootters and W. H. Zurek, “A single quantum cannot be cloned,” *Nature*, vol. 299, pp. 802–803, Oct. 1982, doi: 10.1038/299802a0.
- T. van Leent et al., “Entangling single atoms over 33 km telecom fibre,” *Nature*, vol. 607, no. 7917, pp. 69–73, Jul. 2022, doi: 10.1038/s41586-022-04764-4.
- C. M. Knaut et al., “Entanglement of nanophotonic quantum memory nodes in a telecom network,” *Nature*, vol. 629, no. 8012, pp. 573–578, May 2024, doi: 10.1038/s41586-024-07252-z.
- D. Main et al., “Distributed quantum computing across an optical network link,” *Nature*, vol. 638, no. 8050, pp. 383–388, Feb. 2025, doi: 10.1038/s41586-024-08404-x.
- C. Delle Donne et al., “An operating system for executing applications on quantum network nodes,” *Nature*, vol. 639, no. 8054, pp. 321–328, Mar. 2025, doi: 10.1038/s41586-025-08704-w.
- J. Ang et al., “ARQUIN: Architectures for Multinode Superconducting Quantum Computers,” *ACM Transactions on Quantum Computing*, vol. 5, no. 3, pp. 1–59, Sep. 2024, doi: 10.1145/3674151.
- C. Piveteau and D. Sutter, “Circuit knitting with classical communication,” *IEEE Transactions on Information Theory*, vol. 70, no. 4, pp. 2734 - 2745, Apr. 2025, doi: 10.1109/TIT.2023.3310797.
- S. Gupta, et al. “Gate Teleportation vs. Circuit Cutting in Distributed Quantum Computing,” *Advanced Quantum Technologies* vol. 9, no. 9 (2026). doi: [10.1002/qute.70424](https://doi.org/10.1002/qute.70424).

Acknowledgments

The authors gratefully acknowledge the contributions that were made to this technical report by our colleagues from the University of Chicago. Our sincere appreciation and thanks go to all the individuals who provided insight and expertise that greatly assisted in the research for this paper.

Copyright

Copyright © 2026 memQ, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

memQ reserves the right to change any products described herein at any time, and without notice. memQ assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by memQ. The use or purchase of products does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of memQ. The products described herein may be protected by one or more U.S. patents, foreign patents, or pending applications.

Trademark Information

memQ, the memQ logo, and any marks listed are trademarks of memQ, Inc. Other company and product names may be trademarks of their respective owners.