MUTOWA · PRIVATE EDGE COMPUTE · WHITEPAPER

Private Edge Compute

Millisecond Anycast routing across Canada, the USA, the UK, Japan, and Australia. Zero telemetry. 100% solar. How the network is built, how a request finds its node, and how the Sunlight node program works.

November 2026 · Version 2.1
Winnipeg, Canada · Tokyo, Japan
mutowa.com

Contents

1 Summary 5

1.1 What this edition adds 5

1.2 How to read the figures 5

2 Why the edge, and why private 6

2.1 Distance is latency 6

2.2 Three places to put compute 6

2.3 What the edge is for 7

2.4 Private by structure 7

2.5 Design goals 7

3 The network today 8

3.1 Footprint 8

3.2 Backbone and subsea transits 8

3.3 Beyond fiber: satellite 9

3.4 How the counts are published 9

4 Anycast routing 9

4.1 One address, many places 9

4.2 Health and withdrawal 9

4.3 What a request does, end to end 10

4.4 The latency budget 10

5 The packet path 11

5.1 The network card 11

5.2 Kernel bypass with eBPF XDP 11

5.3 Why eBPF 11

6 Node hardware 12

6.1 Processor 12

6.2 AI accelerator 12

6.3 Edge inference module 12

6.4 Division of labor 13

7 Memory and storage 13

7.1 System memory 13

7.2 The ephemeral RAMDisk 13

7.3 The boot drive . . . 14

8 Confidential computing . . . 14

8.1 AMD SEV-SNP . . . 14

8.2 Attestation . . . 15

8.3 What isolation covers . . . 15

9 The post-quantum mesh . . . 15

9.1 Harvest now, decrypt later . . . 15

9.2 ML-KEM (Kyber-1024) . . . 15

9.3 Hybrid by design . . . 16

9.4 The tunnel and its keys . . . 16

10 Zero telemetry . . . 17

10.1 The rule . . . 17

10.2 Why it is structural . . . 17

10.3 What the rule covers . . . 17

10.4 The protections, taken together . . . 18

11 Power . . . 18

11.1 The power budget of a node . . . 18

11.2 Scheduling with the sun . . . 19

11.3 Redundancy . . . 19

11.4 Siting . . . 19

12 Edge AI and KOTO . . . 19

12.1 AI on the request path . . . 19

12.2 The KOTO ensemble . . . 19

12.3 How the ensemble improves an answer . . . 20

13 Developer interface . . . 20

13.1 Protocols and SDKs . . . 20

13.2 The KOTO API . . . 21

13.3 What an application should assume . . . 21

14 Reliability . . . 21

14.1 What five nines allows . . . 21

14.2 Failure domains . . . 22

14.3 Why stateless nodes fail well . . . 22

14.4 Headroom . . . 22

15 MUTOWA Mobile on the edge 22

16 Sunlight nodes 23

16.1 The four tiers 23

16.2 How a license earns 23

16.3 Reserving, buying and refunds 23

16.4 MUTOWA Flow 24

17 Limits and assumptions 24

18 Glossary 24

19 References and about MUTOWA 25

19.1 Standards and further reading 25

19.2 About MUTOWA 25

19.3 Next steps 26

01

Summary

MUTOWA Private Edge Compute is a network of solar-powered compute nodes that answers each request close to the people who made it. This paper describes how the network is laid out, how a request finds its node, what a node is built from, how traffic and workloads are protected, and how the Sunlight node program lets people hold a place in it.

The network today is 63 compute nodes in 16 cities across 5 countries: Canada, the United States, the United Kingdom, Japan and Australia. Every node carries its own AI accelerator, so inference runs at the edge and not in a distant region. Traffic between nodes is sealed with post-quantum key encapsulation, workloads run inside hardware-isolated virtual machines, and no payload is logged anywhere on the path.

Compute nodes	63
Cities and countries	16 cities in 5 countries
Aggregate bandwidth	16.8 Tbps
Average latency	Under 1.2 ms (P99 round trip across regional fiber links)
Availability target	99.999%
Processor	AMD EPYC Embedded 3251, 8 cores / 16 threads
AI acceleration	One NVIDIA RTX PRO 6000 Blackwell (96 GB GDDR7 ECC) and an NVIDIA Jetson Orin on every node
Memory	128 GB DDR5 ECC, plus a 32 GB ephemeral RAMDisk
Node isolation	AMD SEV-SNP encrypted virtual machines
Mesh encryption	ML-KEM (Kyber-1024) key encapsulation, AES-256-GCM tunnels, keys rotated every 60 seconds
Telemetry	Zero: no payload logging
Power	100% solar, with N+2 UPS redundancy

Table 1. Private Edge Compute at a glance. These are the figures published at mutowa.com/ventures/edge.

1.1 What this edition adds

Version 1 of this paper listed the network's specifications. This edition explains them: why a millisecond budget forces compute to the edge, how Anycast routing and a kernel-bypass packet path work, what hardware isolation does and does not protect, why the mesh uses a post-quantum key exchange today, and how the published availability and power figures translate into operating rules. A chapter on limits and assumptions says plainly what the figures cover.

1.2 How to read the figures

Specifications in this paper are the ones MUTOWA publishes for the network and its nodes. Where the paper works something out from them (a distance from a latency, minutes from a percentage, energy from a power rating), it says so and shows the arithmetic. Live node, city and country counts are published on the Edge page and change as the network grows; the counts printed here are those of November 2026.

02

Why the edge, and why private

Two decisions shape the network. Compute sits near the request because distance is latency, and privacy is built into the hardware and the keys because a promise is not a control.

2.1 Distance is latency

Light in optical fiber travels at roughly two thirds of its speed in a vacuum: about 200,000 km per second, or close to 5 microseconds for every kilometer. A round trip therefore costs about one millisecond for every 100 km of fiber between the two ends, before any router, queue or server has done its work. No protocol removes that cost. The only way to answer in a millisecond is to be within about a hundred kilometers of fiber from whoever is asking.

Fiber path, one way	Round trip in the fiber alone	What it means
50 km	about 0.5 ms	A node in the same metropolitan area
120 km	about 1.2 ms	The published round-trip budget, spent entirely on distance
1,000 km	about 10 ms	A regional data center in another province or state
8,000 km	about 80 ms	A request that crosses an ocean and returns

Table 2. Propagation delay in fiber, at roughly 5 microseconds per kilometer. Arithmetic, not a measurement of the network.

A centralized cloud concentrates capacity in a few very large regions and accepts the distance. That is a reasonable trade for batch work and a poor one for anything interactive: a spoken reply, a camera frame, a control loop, a call. Private Edge Compute makes the opposite trade. It places smaller nodes in more cities and steers each request to the nearest one, so that the round trip between a node and the networks in its region stays under 1.2 ms.

2.2 Three places to put compute

	Centralized cloud region	Content delivery network	Private Edge Compute
Where it runs	A few very large sites	Many points of presence	Compute nodes in 16 cities
Distance to the user	Often hundreds or thousands of kilometers	Short	Short
What runs there	Any workload	Cached content and light functions	Containers and AI models, on every node
AI acceleration	In selected regions	Limited	A GPU accelerator on every node
Who can see a workload's memory	The host, unless confidential computing is used	The host	Encrypted per guest (SEV-SNP)

Table 3. Three ways to place compute, in general terms. The first two columns describe categories, not any one provider.

The comparison shows what is distinctive about the design. Delivery networks already put something near the user, but what they put there is mostly a cache. Private Edge Compute puts a full compute node there, accelerator included, and isolates what runs on it from the node itself.

2.3 What the edge is for

- **Conversation with AI.** An assistant that answers aloud has a budget of a few hundred milliseconds before a pause feels like a delay. Every millisecond not spent on distance is one the model can spend thinking. KOTO is served from the network for this reason (chapter 12).
- **Calls.** Voice and video are continuous and two-way. MUTOWA Mobile is designed to run on the edge network, using nodes near each caller (chapter 15).
- **Places the cloud is far from.** Northern Canadian cities, and with satellite backhaul ships and remote sites, are a long way from any large region (chapter 3).
- **Work that must stay private.** Sensitive inputs are processed in encrypted memory and not kept (chapters 7 to 10).

2.4 Private by structure

Most privacy claims are policies: a statement about what an operator will not do with data it can see. A policy is only as strong as the operator's discipline. Private Edge Compute is designed so that the operator has less to see in the first place.

- **Workloads run in encrypted memory.** Each guest runs in an AMD SEV-SNP virtual machine whose memory is encrypted with a key the host software cannot read (chapter 8).
- **Payloads are never logged.** Nodes keep working data in a RAMDisk that is cleared on every restart; the boot drive holds the operating system image and nothing else (chapter 7).
- **Links are sealed against tomorrow's attacker.** Traffic between nodes is protected with ML-KEM, a key-encapsulation method chosen to resist a quantum computer, and session keys live for 60 seconds (chapter 9).
- **Power is the network's own.** Every node runs on solar power, so the footprint does not depend on a hyperscale hall or its diesel reserve (chapter 11).

2.5 Design goals

Goal	How the design meets it
Answer in about a millisecond	Nodes in 16 cities, Anycast routing to the nearest healthy node, a packet path that bypasses the kernel
Keep payloads unobservable	SEV-SNP guests, ephemeral memory, no payload logging
Stay secure for decades	Post-quantum key encapsulation on the mesh, with a classical exchange alongside it
Bound any compromise	Session keys rotated every 60 seconds and never written to disk
Run AI where the request is	A GPU accelerator and a low-power inference module on every node
Stay up	Anycast failover, no single-node or single-city dependency, N+2 power redundancy
Use clean power only	A fully solar-powered footprint, with scheduled work following the sun

03

The network today

63 nodes in 16 cities and 5 countries, joined by carrier-neutral fiber on land and by trans-Pacific submarine cable.

3.1 Footprint

Country	Cities with nodes	Cities
Canada	Vancouver, Winnipeg, Toronto, Whitehorse, Yellowknife, Iqaluit	6
United States	San Francisco, New York	2
United Kingdom	London	1
Japan	Tokyo, Yokohama, Osaka	3
Australia	Sydney, Brisbane, Canberra, Perth	4
Total	63 compute nodes	16

Table 4. Node cities by country, November 2026. Live counts are published at mutowa.com/ventures/edge.

Canada forms the backbone of the network, with nodes from Vancouver through Winnipeg to Toronto and three northern cities (Whitehorse, Yellowknife and Iqaluit) far from the large data-center regions. The United States provides the coast-to-coast interconnect between San Francisco and New York. Japan lands trans-Pacific capacity and serves the Tokyo, Yokohama and Osaka areas. London and four Australian cities extend the mesh to Europe and Oceania.

3.2 Backbone and subsea transits

Domestic legs run over carrier-neutral fiber. Trans-Pacific capacity rides the FASTER and JUPITER submarine cable systems, which land at Chikura and Shima in Japan and continue to San Francisco and Los Angeles.

Segment	Route
Canada backbone	Vancouver, Winnipeg, Toronto
United States interconnect	San Francisco to New York, coast to coast
Trans-Pacific	FASTER and JUPITER cable landings at Chikura and Shima, to San Francisco and Los Angeles
Japan gateways	Tokyo and Osaka
Peering	Tier-1

Table 5. Published transit segments.

3.3 Beyond fiber: satellite

Satellite backhaul takes Private Edge Compute beyond terrestrial fiber: to ships, remote sites and regions the fiber grid has never reached. MUTOWA announced that satellite access is rolling out in 2026. A satellite hop adds its own propagation delay, so the millisecond figures in this paper describe fiber paths; what satellite adds is reach.

3.4 How the counts are published

The Edge page shows live node, city and country counts taken from the network's own node list, and a map that places every city with a node. The public figures are aggregates. When the live figures cannot be reached, the site falls back to the last published set, which is the one printed in this paper.

04

Anycast routing

Every node answers at the same address. The internet's own routing delivers each request to the nearest one, and takes a node out of the path the moment it stops announcing itself.

4.1 One address, many places

In ordinary (unicast) addressing, an IP address names one place. In Anycast, the same address block is announced to the internet from many places at once. Each network along the way simply forwards a packet toward the announcement that is closest by its own routing metrics. Nothing in the client changes: it resolves one name, connects to one address, and the route it gets is the short one.

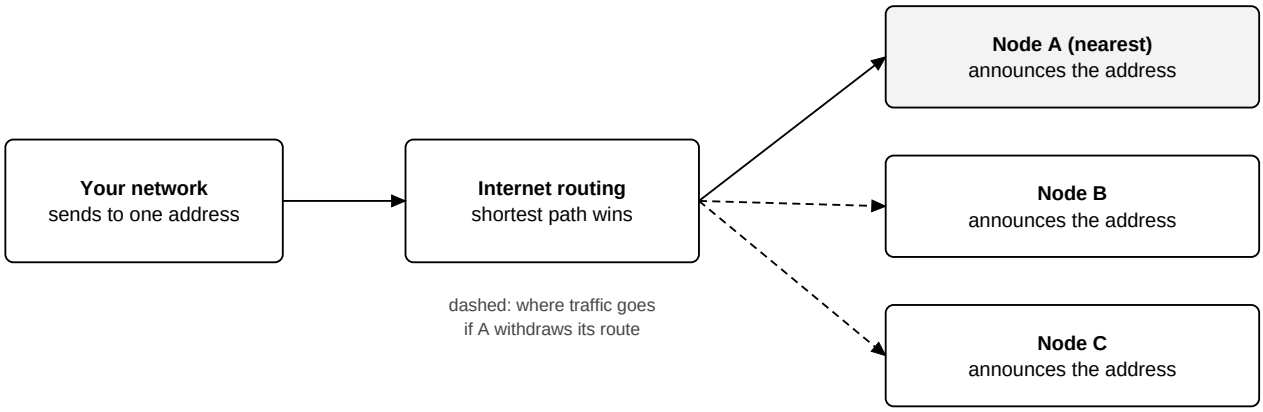


Figure 1. Anycast. Every node announces the same address; routing delivers a request to the nearest. If that node withdraws its announcement, traffic converges on the next.

4.2 Health and withdrawal

A node announces the address only while it is healthy. If a node fails, loses power or is taken out for maintenance, it withdraws its announcement and the routing system converges on the next-nearest node. There is no directory to update and no client to reconfigure. This is the mechanism behind the network's availability target: capacity in one city can drop without anyone being left with an address that no longer answers.

4.3 What a request does, end to end

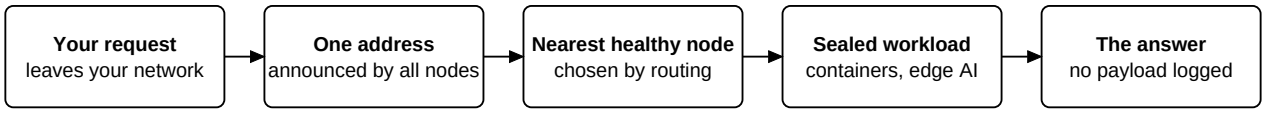


Figure 2. The path of a request across Private Edge Compute.

1. **The same address on every node.** The client sends its request to the network's address.
2. **The nearest node.** Routing delivers the packets to the closest node that is announcing.
3. **The fast path.** The node's network card hands packets to an eBPF XDP program at the driver queue, which routes them in under 40 nanoseconds (chapter 5).
4. **The workload.** A container or an edge-AI model, running in an encrypted virtual machine, does the work.
5. **The answer.** The response returns over the same short path. No payload is logged, and no telemetry leaves the node.

4.4 The latency budget

Average latency	Under 1.2 ms
Node latency	P99 round trip under 1.2 ms across regional fiber links
Packet routing on the node	Under 40 ns (eBPF XDP at the driver queue)
Peering	Tier-1
Protocols	gRPC and WebSocket

Table 6. Published routing figures.

P99 means that 99 of every 100 round trips complete within the stated time; it is a stricter figure than an average because it describes the slow tail and not the typical case. The figure is measured across regional fiber links, that is, between a node and the networks of its own region. A request from farther away still reaches its nearest node, and pays the distance in between at about one millisecond of round trip per 100 km (chapter 2).

> **Design note.** Anycast chooses a node per packet flow, by routing. Long-lived sessions stay on one node for as long as the route is stable; if the route changes mid-session, the session is re-established on the new node. Applications built on the network's gRPC and WebSocket interfaces reconnect in the ordinary way.

05

The packet path

A millisecond budget leaves no room for a slow first hop inside the node itself. The network card and the first piece of software that touches a packet are specified accordingly.

5.1 The network card

Each node has a dual-port 400 GbE controller with QSFP-DD optics on a PCIe 5.0 x16 link. Two ports give a node two independent 400-gigabit paths to the network. The controller supports SR-IOV (single-root I/O virtualization), which presents one physical card as many virtual ones, so that an isolated guest can be given its own network function without sharing a software switch with its neighbors.

5.2 Kernel bypass with eBPF XDP

In a conventional network stack, a packet is copied into a kernel buffer, walked through several protocol layers and queued before any decision is made about it. XDP (the eXpress Data Path) moves the decision to the earliest possible point: a small, verified eBPF program runs inside the network driver, on the receive queue, before the kernel has allocated anything for the packet. The program returns a verdict (drop, pass up the stack, send back out, or redirect to another interface or to a guest) and the driver acts on it at once.

Private Edge Compute routes packets this way in under 40 nanoseconds. For scale, 40 ns is a budget of 25 million routing decisions per second on each processor core, and it is about one thirty-thousandth of the 1.2 ms round-trip figure. The node's own first hop does not appear in the latency a user experiences.

Stage	Where it happens	What it costs
Receive	400 GbE port, hardware queue	Line rate
Route	eBPF XDP program in the driver	Under 40 ns per packet
Deliver	SR-IOV virtual function of the guest, or the host stack	No software switch in the path
Compute	Encrypted guest: container or model	Depends on the workload

Table 7. The stages a packet crosses inside a node.

5.3 Why eBPF

An eBPF program is checked by the kernel's verifier before it is allowed to run: it must terminate, stay within its memory and use only permitted helpers. That gives the fast path the speed of code running in the driver with a safety check a hand-written driver patch would not have, and it lets routing logic be updated on a running node without rebooting it.

Network card	Dual-port 400 GbE QSFP-DD, PCIe 5.0 x16, SR-IOV
Kernel bypass	eBPF XDP direct driver-queue packet routing in under 40 ns
Aggregate bandwidth	16.8 Tbps across the network

Table 8. Published network hardware.

06

Node hardware

A node is a small, sealed computer: an embedded server processor, a workstation-class AI accelerator, a low-power inference module, error-correcting memory and a fast network card, in an aluminum chassis.

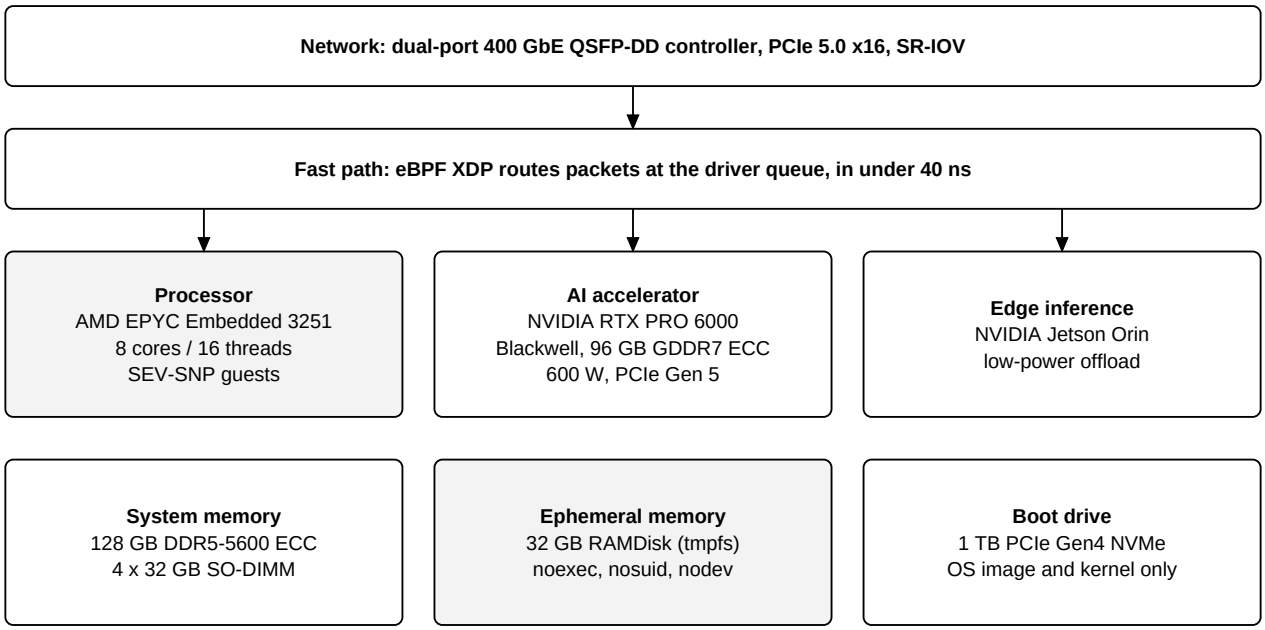


Figure 3. Inside a node. Packets arrive at the network card, are routed in the driver and reach the processor, the accelerator or the inference module. The bottom row is what a node holds: working data lives only in memory.

6.1 Processor

The system-on-chip is an AMD EPYC Embedded 3251: 8 cores and 16 threads, 2.5 GHz base and 3.1 GHz boost. It sits in a passively cooled, fanless aluminum chassis with a configurable thermal design power of 15 W to 55 W. An embedded part is chosen over a larger server processor for three reasons: it runs within a solar power budget, it needs no fan (so the processor's cooling has no moving part to fail), and it carries the memory-encryption hardware that node isolation depends on.

6.2 AI accelerator

Every node carries one NVIDIA RTX PRO 6000 Blackwell: 24,064 CUDA cores, 96 GB of GDDR7 error-correcting memory, PCIe Gen 5, and a 600 W thermal design power with dual flow-through active cooling. The 96 GB of accelerator memory is what allows large models to be held on a single node, with error correction protecting long-running inference from memory faults.

6.3 Edge inference module

Beside the accelerator sits an NVIDIA Jetson Orin, for low-power inference offload. Small models and always-on tasks run there at a fraction of the accelerator's power, which keeps a node useful when its power budget is tight and leaves the large accelerator for work that needs it.

Part	Specification	Role
Processor	AMD EPYC Embedded 3251: 8 cores / 16 threads, 2.5 GHz base, 3.1 GHz boost	Routing, orchestration, encrypted guests
AI accelerator	1 x NVIDIA RTX PRO 6000 Blackwell: 24,064 CUDA, 96 GB GDDR7 ECC, 600 W TDP, PCIe Gen 5	Large-model inference and generation
Edge inference	NVIDIA Jetson Orin	Low-power inference offload
Isolation	AMD SEV-SNP encrypted virtualization with PSP-managed memory keys	Keeps guest memory unreadable to the host
Thermal design	Processor: fanless aluminum chassis, 15 W to 55 W configurable TDP. Accelerator: dual flow-through active cooling at 600 W	Runs inside a solar budget
Chassis	Aluminum, with cooling fins down each side	Passive heat path for the processor

Table 9. Compute hardware of a node.

6.4 Division of labor

The three compute parts are not redundant; each takes the work it is most efficient at. The processor routes, schedules and hosts the isolated guests. The inference module answers small, frequent requests at low power. The accelerator takes the requests that need its memory and throughput. The design lets a request be served by the least power-hungry part that can serve it well.

07

Memory and storage

What a node can remember decides what it can leak. The storage design gives a node a large, fast working memory and almost nothing that persists.

7.1 System memory

A node has 128 GB of DDR5-5600 memory in four 32 GB unbuffered SO-DIMMs, with error-correcting code (ECC). ECC memory detects and corrects single-bit faults as they occur. On a machine that runs unattended for long periods, that is the difference between a silent corruption and a corrected, counted event.

7.2 The ephemeral RAMDisk

Working data (request payloads, intermediate results, model inputs and outputs) lives in a 32 GB RAMDisk: a tmpfs file system held entirely in memory. It never touches a disk, and its contents are gone the moment the node restarts or loses power. The RAMDisk is mounted with three restrictions:

Mount option	What it does	What it prevents
noexec	Nothing stored on the RAMDisk can be executed	A payload being run as a program
nosuid	Set-user-ID bits are ignored	A file being used to gain privileges
nodev	Device files are not honored	A file being used to reach hardware directly

Table 10. The RAMDisk's mount options.

7.3 The boot drive

The only persistent storage is a 1 TB Samsung 990 Pro PCIe Gen4 NVMe drive, which holds the operating system image and the kernel. It is where a node starts from, not where it works.

Kind of data	Where it lives	How long it lasts
Operating system image, kernel	Boot drive (NVMe)	Until the next software release
Request payloads, working files	32 GB RAMDisk (tmpfs)	Until the work is done; never past a restart
Guest memory	System RAM, encrypted per guest (SEV-SNP)	While the guest runs
Session keys	Memory only	60 seconds

Table 11. What persists on a node, and what does not.

08

Confidential computing

A workload on Private Edge Compute runs in a virtual machine whose memory the node's own host software cannot read. That property comes from the processor, not from a policy.

8.1 AMD SEV-SNP

Secure Encrypted Virtualization (SEV) encrypts the memory of each virtual machine with its own key. The keys are generated and held by a separate security processor inside the chip (the AMD Platform Security Processor, PSP) and are never exposed to the hypervisor or the host operating system. Someone with full control of the host sees only ciphertext when they look at a guest's memory.

Secure Nested Paging (SNP) adds integrity to that confidentiality. The processor tracks which guest owns each page of memory and refuses writes or remappings that the owner did not make. That closes a class of attacks in which a malicious host does not read a guest's memory but replays, swaps or corrupts it to change what the guest computes.

8.2 Attestation

An SEV-SNP guest can ask the processor for a signed report that states what was loaded into it and on what hardware and firmware. The report is signed with a key rooted in the processor itself, so a remote party can check that it is talking to a genuine isolated guest running the expected software before it sends anything sensitive.

8.3 What isolation covers

Threat	Covered by SEV-SNP	Notes
Host software reads guest memory	Yes	Memory is encrypted with a key the host never holds
Host remaps or replays guest memory	Yes	Integrity checks reject pages the guest did not write
Another guest reads this guest	Yes	Each guest has its own key
A flaw in the guest's own software	No	Isolation protects a workload from its host, not from itself
Traffic observed in transit	No	Covered by the encrypted mesh (chapter 9)

Table 12. What hardware isolation does and does not protect.

Isolation, ephemeral memory and encrypted links are three separate layers that cover each other's gaps: data is encrypted while it moves, encrypted while it is used, and not kept once it has been used.

09

The post-quantum mesh

Traffic between nodes is sealed today against an attacker who does not exist yet: one with a quantum computer large enough to break the public-key cryptography the internet runs on.

9.1 Harvest now, decrypt later

Encrypted traffic can be recorded and stored. Whoever records it cannot read it now, but if the key exchange that protected it is ever broken, everything recorded becomes readable at once. The key exchanges in common use (elliptic-curve Diffie-Hellman among them) rest on mathematical problems that a sufficiently large quantum computer would solve with Shor's algorithm. Data that must stay private for ten or twenty years is therefore exposed by a machine that might be built within that time, even though no such machine exists today. The defense has to be in place before the recording, not before the machine.

9.2 ML-KEM (Kyber-1024)

The mesh establishes its keys with ML-KEM, the module-lattice key-encapsulation mechanism that NIST standardized as FIPS 203 in August 2024 from the CRYSTALS-Kyber design. Its security rests on the hardness of a lattice problem (module learning with errors) for which no efficient quantum algorithm is known. The network uses the largest parameter set, ML-KEM-1024, which targets NIST security category 5: at least as hard to break as AES-256.

Property	ML-KEM-1024	X25519, for comparison
Kind	Key encapsulation (lattice-based)	Key agreement (elliptic curve)
Public key	1,568 bytes	32 bytes
Ciphertext or key share	1,568 bytes	32 bytes
Shared secret	32 bytes	32 bytes
Standard	NIST FIPS 203	RFC 7748
Resists a quantum computer	Designed to	No

Table 13. Parameter sizes from the public standards.

9.3 Hybrid by design

Lattice cryptography is newer than elliptic curves and has had fewer years of attack. The mesh therefore runs both: a classical X25519 exchange and an ML-KEM-1024 encapsulation, with the session key derived from the two secrets together. An attacker has to break both to recover the key. If a flaw were found in the lattice scheme, the traffic would be as safe as it is on today's internet; if a quantum computer arrives, the lattice half still holds.

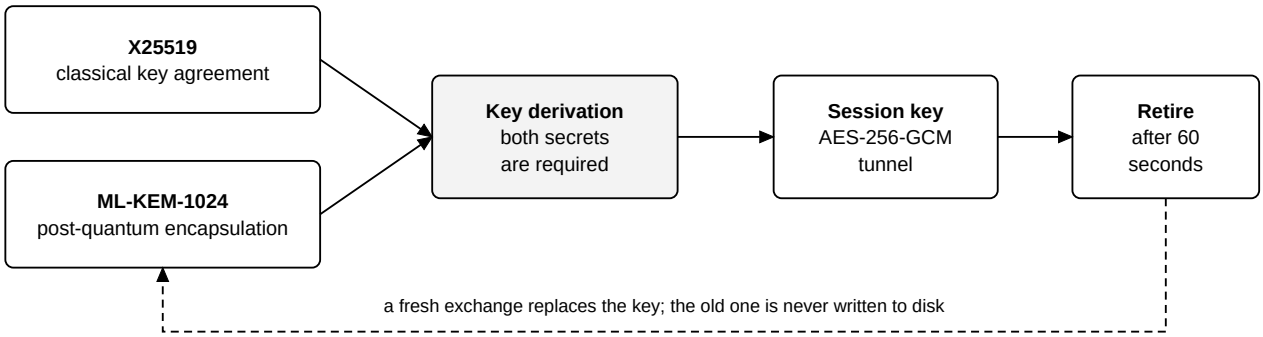


Figure 4. Hybrid key establishment and rotation on the mesh.

9.4 The tunnel and its keys

Once a session key exists, traffic is carried in WireGuard tunnels sealed with AES-256-GCM, an authenticated cipher: it keeps the contents secret and detects any modification. Symmetric ciphers are far less exposed to quantum attack than key exchanges are. The best known quantum attack (Grover's algorithm) roughly halves the effective key length, which leaves a 256-bit key with a margin comparable to 128 bits against a classical attacker.

Session keys rotate every 60 seconds, from random numbers produced by the processor's true random number generator, and are never written to disk. Rotation gives the mesh forward secrecy with a short window: a key that leaked would expose at most one minute of one link's traffic, and there are 1,440 independent keys in a day.

Key encapsulation	NIST ML-KEM (Kyber-1024)
Transport	Hybrid X25519 + Kyber-1024; AES-256-GCM WireGuard tunnels
Key rotation	Every 60 seconds, from the CPU's true random number generator
Forward secrecy window	60 seconds

Table 14. Published mesh security figures.

10

Zero telemetry

Zero telemetry is an operating rule with hardware behind it: no payload is logged at any point on the path, and a node keeps nothing that would let a payload be reconstructed later.

10.1 The rule

- **No payload logging.** The content of a request and its answer is not written to any log, on the node or off it.
- **Ephemeral memory only.** Working data sits in the RAMDisk and disappears on restart.
- **Encrypted in use.** Guest memory is encrypted by the processor, so the host cannot observe it.
- **Short-lived keys.** Session keys last 60 seconds and are never stored, so recorded traffic cannot be opened afterward with a key taken from a node.

10.2 Why it is structural

Each part of the rule is enforced by something other than intent. There is no payload log because there is nowhere persistent to write one except the boot drive, which holds the system image. The host cannot inspect a guest because it does not hold the guest's memory key. A stored recording cannot be decrypted from a seized node because the keys that protected it expired within a minute and were never on disk.

10.3 What the rule covers

The rule is about payloads: what a request says and what the answer contains. Like every network, the path has to read a packet's addressing in order to deliver it, and a node has to know whether it is healthy in order to announce itself. Those are properties of routing, not records of what people sent. MUTOWA's published commitment is that payloads are never logged and that node state is ephemeral.

10.4 The protections, taken together

Who or what	What they could try	What stands in the way
Someone recording a link	Store traffic to read later	Hybrid key exchange, AES-256-GCM, keys that last 60 seconds
A future quantum attacker	Break the key exchange of recorded traffic	ML-KEM-1024 alongside X25519
Someone altering traffic	Inject or modify packets on a link	Authenticated encryption rejects anything changed
Host software on a node	Read or tamper with a guest's memory	SEV-SNP: memory encrypted and integrity-checked per guest
Someone who takes a node	Read what is stored on it	Only the system image persists; working data and keys were in memory
A failed node or site	Leave users with a dead address	Anycast withdrawal moves traffic to the next node
A flaw in a workload itself	Leak its own data	Not covered: this is the application's responsibility

Table 15. The network's protections against each kind of threat, from chapters 4 and 7 to 10.

> **Scope.** The public website counts page views with its own first-party counter, which stores nothing in the visitor's browser and loads no third-party script. That is a property of mutowa.com, separate from the zero-telemetry rule for workloads on the network.

11

Power

The whole footprint runs on solar power. That is a constraint the hardware and the scheduler are designed around, not a certificate bought afterward.

11.1 The power budget of a node

Part	Thermal design power	Energy at that power
Processor, low setting	15 W	0.36 kWh per day
Processor, high setting	55 W	1.32 kWh per day
AI accelerator at full load	600 W	0.6 kWh per hour

Table 16. Published power ratings, with the energy they imply. Thermal design power is the heat a part is built to shed at sustained full load; typical draw is lower.

The arithmetic explains the architecture. A node's always-on parts (the processor, memory and network card, and the low-power inference module) draw tens of watts and can run around the clock from a modest array and its battery. The accelerator draws an order of magnitude more, so heavy work is what the scheduler moves in time.

11.2 Scheduling with the sun

KOTO, the network's flagship workload, is run by a solar-aware load director. Interactive work (chat and tool calls) is served first, from live solar output and the battery reserve. Background jobs and fine-tuning are scheduled against the real-time insolation curve, so the most power-hungry work runs when the panels produce the most. Above 750 W/m² of irradiance, every KOTO node runs at peak throughput.

11.3 Redundancy

Nodes have N+2 UPS power redundancy: two more uninterruptible supply units than the load needs, so that one can fail while another is being serviced without the node losing power. Combined with Anycast, a site that does lose power withdraws its route and its traffic moves to the next node (chapter 14).

11.4 Siting

Sites are chosen for sunlight and connectivity together: solar capacity in each country, tied to Tier-1 fiber and, on the Pacific, to the cable landings. Clean power and millisecond routing are then properties of the same geography.

12

Edge AI and KOTO

Every node can run models, so AI on the network is served on the same short path as everything else. KOTO, the assistant made by KOTO, a MUTOWA company, is the workload the network was sized for.

12.1 AI on the request path

In a centralized design, a request travels to an application region and then again to wherever the GPUs are. On Private Edge Compute the accelerator is in the node that received the request. Inference is one more stage on the node (chapter 5), inside the same encrypted guest, under the same no-logging rule. There is no separate AI region and no second network hop.

12.2 The KOTO ensemble

KOTO is not one large model but an ensemble: 42 specialist models of 32 billion parameters each, 1.34 trillion parameters in total, coordinated in real time on every request. Reasoning, code, vision, language, file and retrieval specialists work in parallel and their results are assembled and verified before an answer is returned.

Model	Context window	Output
Flash 1.2	128K tokens	8K tokens
Super 1.2	1M tokens	1M tokens
Super Heavy 2	2M tokens	2M tokens

Table 17. KOTO models and their context windows.

Ensemble	42 specialist models of 32B parameters
Aggregate parameters	1.34T
Throughput	About 200 tokens per second, sustained
Availability target	99.999%
Modalities	Text, code, files and images in and out; audio in
In transit / in use	ML-KEM (Kyber-1024) / AMD SEV-SNP
Telemetry	Zero

Table 18. The KOTO spec sheet, as published at mutowa.com/ventures/ai.

12.3 How the ensemble improves an answer

- **Self-consistency sampling.** KOTO produces one deterministic answer and several independent samples; the majority answer wins. Without consensus it adds samples, raises the thinking budget and tightens answer extraction.
- **Long context.** Conversations that outgrow the window are summarized from the oldest turn; the latest request is always kept intact. Long answers extend across automatic continuations.
- **Solar-aware scheduling.** Interactive requests are served first; batch work follows the sun (chapter 11).

Full model documentation is at api.mutowa.com/docs.

13

Developer interface

Applications reach the network through two protocols and three SDKs, with endpoints in all five countries.

13.1 Protocols and SDKs

The network speaks gRPC for request and response calls and WebSocket for long-lived, two-way streams. SDKs are provided for TypeScript, Go and Rust. Because the address is Anycast, a client is configured once and reaches its nearest node wherever it runs.

Protocols	gRPC and WebSocket
SDKs	TypeScript, Go, Rust
Peering	Tier-1
KOTO API base	api.mutowa.com
KOTO wire formats	Chat completions and messages
Streaming	Server-sent events (SSE)

Table 19. The developer interface.

13.2 The KOTO API

KOTO is served through an API that accepts the two request formats most applications already use (chat completions and messages) and streams responses as server-sent events. Three models cover the range from high-throughput work (Flash 1.2) through general assistant tasks (Super 1.2) to deep reasoning over very large contexts (Super Heavy 2).

13.3 What an application should assume

- A request is answered by the nearest healthy node; which node that is can change.
- State that must outlive a request belongs in the application, not on the node: node memory is ephemeral by design.
- Traffic between nodes is sealed on the post-quantum mesh, and workloads run in encrypted guests.
- Long-lived streams should reconnect on interruption, as on any network.

14

Reliability

The network's availability target is 99.999%. This chapter explains what that number means and what in the design supports it.

14.1 What five nines allows

Availability	Downtime per year	Downtime per month
99.9%	8 hours 46 minutes	about 44 minutes
99.99%	53 minutes	about 4 minutes
99.999%	5 minutes 15 seconds	about 26 seconds

Table 20. What an availability percentage permits. Arithmetic on a 365-day year.

Five minutes a year cannot be met by repairing things quickly; a single reboot would spend the budget. It can only be approached by making the failure of any one part invisible to the user, which is what Anycast failover and the absence of persistent node state are for.

14.2 Failure domains

What fails	What happens	What the user sees
One node	It withdraws its announcement; routing converges on the next node in the city or region	A reconnect at most
One city	All of its nodes withdraw; traffic goes to the next-nearest city	Higher latency, by the added distance
One fiber route	Routing converges on the remaining paths	A brief reroute
Power at a site	N+2 UPS carries the load; if the site still goes dark it withdraws its route	Nothing, or a reconnect
A session key	It expires within 60 seconds and is replaced	Nothing

Table 21. Failure domains and how the design contains them.

14.3 Why stateless nodes fail well

Because a node keeps no payloads and no long-lived keys, there is nothing to recover when one fails and nothing to migrate when one is replaced. A restarted node boots its image, joins the mesh with fresh keys and starts announcing. The properties that make the network private are the same ones that make it easy to keep up.

14.4 Headroom

The network carries 16.8 Tbps of aggregate bandwidth. Headroom matters to availability because failover moves traffic: when a city drops out, its neighbors must be able to absorb its load without slowing down.

15

MUTOWA Mobile on the edge

MUTOWA Mobile is cell service that runs on the MUTOWA Edge network. Its design is published as MUTOWA Standard 0926-PQTP, the Private Quantum-Edge Telephony Protocol.

A call is the hardest test of an edge network: it is interactive, it is continuous and both ends notice every lost packet. In the standard, a node may do what it is placed to do: resolve a number to a device, relay a call when two phones cannot reach each other directly, and screen unknown callers inside a confidential-compute environment. Calls are designed to be protected with the same family of cryptography as the mesh: X25519 together with ML-KEM-1024, with ML-DSA-87 device signatures.

Three things have been announced, and nothing more: early previews are rolling out to Sunlight node license holders as the network expands, MUTOWA numbers are coming soon, and MUTOWA plans for data plans to be completely free. The plan for free data is a plan, not yet an offer.

The design is described in full in the companion paper, *Private Quantum-Edge Telephony*, available at mutowa.com/press.

16

Sunlight nodes

A Sunlight node is a place in the network, hosted by MUTOWA or shipped to its holder to plug in and connect, as the holder chooses. The holder is compensated for the usage the node's tier serves, and a node earns only while it is online.

16.1 The four tiers

Tier	Price (USD)	Usage it is paid for	Includes
Lite node	$100	Network	A smaller share of network capacity than the Network node
Network node	$500	Network	A full network node slot on Private Edge Compute
Compute node	$1,500	Network and compute	Compute capacity for Private Edge Compute workloads; 10 TB of DREAM cloud storage, plus 1 TB per verified referral; priority access. Limited to 1,000 licenses
Quantum node	$2,500	Network, compute and quantum compute	Everything in Compute, and a small QPU allocation for quantum compute workloads

Table 22. Sunlight node tiers, as published at mutowa.com/nodes.

16.2 How a license earns

Compensation comes from real usage of the streams a tier includes: network usage, compute usage and quantum compute usage. Each online node's share is recorded every day. Nothing accrues while a node is offline.

> **Earnings.** Compensation depends on real usage of the streams a tier includes and on node uptime. Earnings vary and are not guaranteed.

16.3 Reserving, buying and refunds

- **Reserving is free.** Choosing a tier and leaving an email address charges nothing. The listed prices are license prices, not reservation charges.
- **Activation comes in waves**, announced by email. The final service terms and availability should be reviewed before deciding to purchase.
- **The Compute tier is capped** at 1,000 licenses.
- **Referrals.** After reserving, a holder can share a link; referrers can earn up to $2,000 from verified referrals.
- **Refunds.** A license bought in MUTOWA Flow can be refunded to the holder's Flow balance for as long as its node has never been online. Once a node has been online, its license is not refundable. The terms of sale are at mutowa.com/sales.

16.4 MUTOWA Flow

License holders follow their nodes in MUTOWA Flow: each node's status, its recorded usage, the balance and payouts, in one place. Signing in takes a password and a code from an authenticator app, with recovery codes as a backup. Payouts are requested from Flow by bank transfer, crypto or Interac e-Transfer (Canada); each request is reviewed and then paid or declined.

17

Limits and assumptions

A specification is only useful if its edges are stated. This chapter says what the figures in this paper cover, and what they do not.

- **Latency figures describe regional fiber paths.** Under 1.2 ms is the round trip between a node and the networks of its region. Distance beyond that adds delay that no network can remove (chapter 2).
- **Availability is a target.** 99.999% is what the network is designed and operated to meet. It is not a service-level agreement; any commitment to a customer is set out in that customer's service terms.
- **Counts change.** Node, city and country counts are those of November 2026. The live figures are on the Edge page.
- **Isolation protects a workload from its host.** It does not protect a workload from a flaw in its own software, and it does not replace the application's own access control.
- **Post-quantum means designed to resist.** ML-KEM is a public standard that has withstood years of open analysis; no cryptography is proven unbreakable, which is why the mesh pairs it with a classical exchange.
- **Zero telemetry concerns payloads.** Routing necessarily handles addressing in transit (chapter 10).
- **Thermal design power is a ceiling, not a meter.** The energy figures in chapter 11 are arithmetic on published ratings.
- **Satellite extends reach, not speed.** A satellite hop adds delay.
- **Specifications are subject to change** as the network evolves.

18

Glossary

Term	Meaning
AES-256-GCM	A symmetric cipher with a 256-bit key that both encrypts data and detects tampering
Anycast	Announcing one address from many places, so routing delivers each packet to the nearest
Attestation	A signed statement from the processor about what software an isolated guest is running
eBPF	Small verified programs that run inside the operating system kernel
ECC memory	Memory that detects and corrects bit errors
Forward secrecy	The property that a key compromised today does not expose earlier traffic

Term	Meaning
Harvest now, decrypt later	Recording encrypted traffic today to decrypt it when the key exchange can be broken
Hybrid key exchange	Deriving a key from a classical and a post-quantum exchange together
ML-KEM	The lattice-based key-encapsulation standard (NIST FIPS 203), from CRYSTALS-Kyber
P99	The value that 99 of every 100 measurements fall within
PSP	AMD Platform Security Processor: the chip's separate security processor, which holds memory keys
RAMDisk (tmpfs)	A file system held in memory, cleared on restart
SEV-SNP	AMD's encrypted virtualization with integrity protection for guest memory
SR-IOV	Presenting one physical network card as several virtual ones
Sunlight node	A place in the MUTOWA network, hosted by MUTOWA or shipped to its holder
TDP	Thermal design power: the heat a part is built to shed at sustained load
XDP	The eXpress Data Path: running eBPF on packets in the network driver

19

References and about MUTOWA

19.1 Standards and further reading

1 NIST. *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard.* August 2024.

2 NIST. *SP 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.* 2007.

3 A. Langley, M. Hamburg, S. Turner. *RFC 7748: Elliptic Curves for Security.* IETF, 2016.

4 J. Abley, K. Lindqvist. *RFC 4786: Operation of Anycast Services.* IETF, 2006.

5 T. Høiland-Jørgensen et al. *The eXpress Data Path: Fast Programmable Packet Processing in the Operating System Kernel.* ACM CoNEXT, 2018.

6 AMD. *AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More.* White paper, 2020.

7 P. W. Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.* SIAM Journal on Computing 26, 1997.

8 MUTOWA. *Quantum Compute* and *Private Quantum-Edge Telephony.* Whitepapers, November 2026. mutowa.com/press.

19.2 About MUTOWA

Rooted in the philosophy of **無永遠** (MUTOWA), the Japanese concept of boundless permanence, MUTOWA builds enduring, fault-tolerant infrastructure for the modern world. MUTOWA develops a connected ecosystem of edge intelligence, applied AI and quantum-ready systems designed for the long horizon. KOTO, a MUTOWA company, makes the KOTO assistant and the KOTO devices.

MUTOWA is led by its co-founders, Nicolai Kazuyuki Diaz Tsuzuki (Co-Founder and President) and Parker Czuba (Co-Founder and Chief Technology Director), from Winnipeg, Canada and Tokyo, Japan.

19.3 Next steps

- See the live network: mutowa.com/ventures/edge
- Compare and reserve a Sunlight node: mutowa.com/nodes
- Read the KOTO documentation: api.mutowa.com/docs
- Write to MUTOWA: hello@mutowa.net, or investors@mutowa.net about nodes

© 2026 MUTOWA. Specifications are subject to change as the network evolves. Earnings from a Sunlight node vary and are not guaranteed.