BLUEDOT

COMPETITIVE BAKE-OFF

MikroTik vs Quantum

An IPoE BNG Bake-Off — Session Architecture, Embedded AAA and Billing Data

A round-by-round comparison of a MikroTik RouterOS device configured as an IPoE BNG against the Quantum IPoE BNG Platform Community Edition with its embedded AAA and embedded DHCP. The comparison turns on one thing: what each product keeps in its session table, and what that makes possible — above all, the per-subscriber throughput data that gives Quantum's embedded AAA its billing detail.

Document Type	BlueDot Competitive Bake-Off
Version	**v8** — supersedes v7
Contenders	RouterOS 7 + `user-manager` vs Quantum IPoE BNG CE 1.4
Audience	Operators, Field, Product
Classification	**BlueDot Proprietary**
Date	November 2026

BlueDot Insight · bluedot-insight.com · © 2026 BlueDot · All rights reserved.

Contents

Executive Summary 4
The Bake-Off 5
The scorecard, up front 6
What we compared, and how to read it 6
Round 1 — What Arrives When You Download 7
The shopping list 7
The glue you write yourself 8
Day one, hour one 8
Round 2 — The Session Table 10
Where the subscriber lives 10
MikroTik: three tables that never join 10
Quantum: one record, and it is the join 11
What the session table makes possible 12
Round 3 — Throughput, Subscriber by Subscriber 14
Two outcomes versus six 14
The measurement chain, end to end 15
Reading it at any cadence 15
Precision 16
Round 4 — The Billing Data 18
What each product hands the billing system 18
The Quantum record, field by field 19
Worked example — one subscriber, one month 19
When the billing consumer goes away 21
Round 5 — Changing a Live Subscriber 22
MikroTik on IPoE: the lease is not a session 22
Quantum: one record write, live effect, reported back 22
Round 6 — Who the Subscriber Is 25
MikroTik: the MAC is the identity 25
Quantum: identity, and the locators that find it 25

Round 7 — Provisioning, and What Survives an Upgrade 27
Provisioning 27
The question nobody asks until it bites 27
Round 8 — Scale, Licensing and Operating Cost 29
Round 9 — Where MikroTik Wins 30
The Scorecard 32
Session model and identity 32
Measurement and billing 33
Control of a live subscriber 34
Platform and operations 35
Breadth outside IPoE 35
How to Decide 36
The one-paragraph version 36
Appendix A — MikroTik Reference 37
Packaging and lineage 37
Documented absences 37
RFC posture 38
Messaging channels 38
The DHCP seam 39
Vendor attribute surface — PEN 14988 41
Component structure 44
Appendix B — Quantum Reference 45
The local AAA decision 45
The REST surface 46
Defaults that matter operationally 47
Terminate causes — RFC 2866 numerics 47
Onboarding call flow 48
Where Quantum goes next 48
Sources 50

Executive Summary

For a dedicated IPoE BNG, Quantum is the winner — and the reason is architectural, not a feature count.

A MikroTik device configured as an IPoE BNG is not a BNG. It is a **multi-component solution the operator hand-integrates**: RouterOS for forwarding, a separately downloaded `user-manager` package for AAA, a DHCP server wired to it over RADIUS, and a simple-queue topology the operator designs for rate enforcement. Those parts were built for different jobs and never fully join, so a subscriber ends up spread across **three tables with three keys and three lifetimes** — a lease with no counters, an accounting row with two totals, and a queue re-zeroed whenever the rate changes.

Quantum is one product, built for one job. DHCP and AAA are embedded, and the subscriber is **one IP-keyed session record** carrying identity, plan, address and a handle to twelve per-session counters in the data plane. That is why its embedded AAA emits a conformance-resolved billing record per subscriber, why a tier change reaches the policer immediately rather than at the next lease renewal, and why the platform is carrying commercial IPoE traffic within an hour of download — with no RADIUS server in the path.

The nine rounds that follow test that claim one capability at a time.

The Bake-Off

Two boxes. The same job: terminate IPoE subscribers, authorise them, rate-limit them, and produce the data an ISP bills from.

	Contender A	Contender B
Product	MikroTik RouterOS 7	**Quantum IPoE BNG Platform**
Forwarding	RouterOS	VPP / DPDK
DHCP	RouterOS DHCP server	**Embedded**
AAA	`user-manager` package (separate download)	**Embedded local AAA**
Subscriber database	SQLite on device flash	QuantumTouch, system of record
Enforcement	Dynamic simple queues	RFC 2698 trTCM policer, per session
Edition compared	RouterOS 7 current	**Community Edition 1.4 — free**

Both can hand out an address and cap a speed. That is where the similarity ends, and the reason is not a feature list. It is **what each product keeps in its session table.**

MikroTik spreads a subscriber across three unrelated tables with three different keys and three different lifetimes. Quantum keeps one IP-keyed session record that holds identity, policy, address, usage and live throughput together. Every round in this document is downstream of that single difference.

The scorecard, up front

Round	What is being compared	Winner
1	What arrives when you download	**Quantum** — complete and running in under an hour
2	**The session table**	**Quantum** — one record vs three disjoint tables
3	Per-subscriber throughput data	**Quantum** — 12 band counters vs two totals
4	Billing data quality	**Quantum** — conformance-grade CDRs, durable spool
5	Changing a live subscriber	**Quantum** — MikroTik cannot, on IPoE
6	Who the subscriber is	**Quantum** — identity separate from locators
7	Provisioning, and surviving an upgrade	**Quantum** — REST + external system of record
8	Scale and licensing	**Quantum** — 5,000 free; 100,000 commercial
9	Breadth outside IPoE	**MikroTik** — EAP, RadSec, arbitrary VSAs, prepaid calendars

Eight rounds to one. Round 9 is real and we state it plainly: if the question is "*can one box be my RADIUS server for Wi-Fi, hotspot, PPPoE and IPoE at once?*", MikroTik answers better. If the question is "*can I run an IPoE subscriber base and bill it properly?*", only one of the two answers at all.

What we compared, and how to read it

Contender A is a RouterOS device configured the way MikroTik documents an IPoE deployment: a DHCP server with `use-radius=yes`, pointed at the `user-manager` package on the same box or a neighbouring one, with rate limits delivered as `Mikrotik-Rate-Limit` and enforced by dynamic simple queues. Everything claimed about Contender A is drawn from MikroTik's own published documentation, with the field record used only where the documentation is silent.

Contender B is the Quantum IPoE BNG Platform Community Edition, release 1.4 — the free download. Everything claimed about Contender B is drawn from its shipped source and release notes. Where a capability lands in 1.5 rather than 1.4, it is marked **[1.5]** at the point of use.

Round 1 — What Arrives When You Download

A bake-off that starts after installation hides half the result. So start at the download.

The shopping list

To run IPoE with per-subscriber authorisation, rate control and usage billing, an operator needs seven things. Here is where each one comes from.

Needed	MikroTik	Quantum CE 1.4
BNG forwarding	RouterOS — in the box	Quantum BNG (VPP/DPDK) — in the box
DHCP server	RouterOS — in the box	**Embedded** — in the box
AAA / RADIUS server	`user-manager` — separate package, download and install	**Embedded local AAA** — in the box
Subscriber database	User Manager's SQLite on flash	QuantumTouch `aaa.db` — in the box
AAA management UI	**None.** Removed in RouterOS 7 — CLI, Winbox or WebFig	QuantumTouch console — in the box
Per-subscriber rate enforcement	Dynamic simple queues you must lay out	trTCM policer, programmed per session
Usage / billing export	CLI, Winbox or the `/um` end-user page	Versioned REST, cursor + ack

Quantum's row reads "in the box" seven times. That is the product claim, and it is a literal one: **an operator downloads the platform, runs the setup wizard, and carries commercial IPoE traffic in under an hour — with no RADIUS server, no FreeRADIUS, and no third-party software anywhere in the subscriber path.**

The embedded AAA is not a stepping stone to a real RADIUS server. It is the AAA service, fully featured, and for a standalone deployment it is all an operator needs. An external RADIUS connector exists as an *optional* future integration for operators who already run a RADIUS estate and want Quantum to join it — not as a gap being filled.

The glue you write yourself

The table above lists components. The difference in effort is in what is *not* a component.

On MikroTik you design and maintain:

- A queue topology — parent queues, `insert-queue-before` ordering, and a convention that keeps dynamic per-lease queues from colliding with static ones.
- An Option 82 strategy. MikroTik's own insertion sets circuit-ID to the ingress interface MAC and remote-ID to the client MAC, which carry no line identity, so the useful case is Option 82 arriving from an access node upstream.
- An accounting consumer. The session table is readable from the CLI and the `/um` page; turning it into invoices is a scripting exercise against a device.
- A provisioning path. One user per CLI invocation, with no AAA API.
- A backup discipline, because the subscriber database is a SQLite file on the device's flash and there is no replication, no HA and no external backend.

On Quantum, for the standard path, you write none of it. The policer is programmed from the plan at onboard. Line identity is carried on the session as `circuit_id` and `remote_id` and matched to an account. Billing records drain over a documented cursor API. Provisioning is REST, atomic and bulk-capable to 10,000 records. And the subscriber database lives in QuantumTouch, deliberately outside the BNG — Round 7.

Day one, hour one

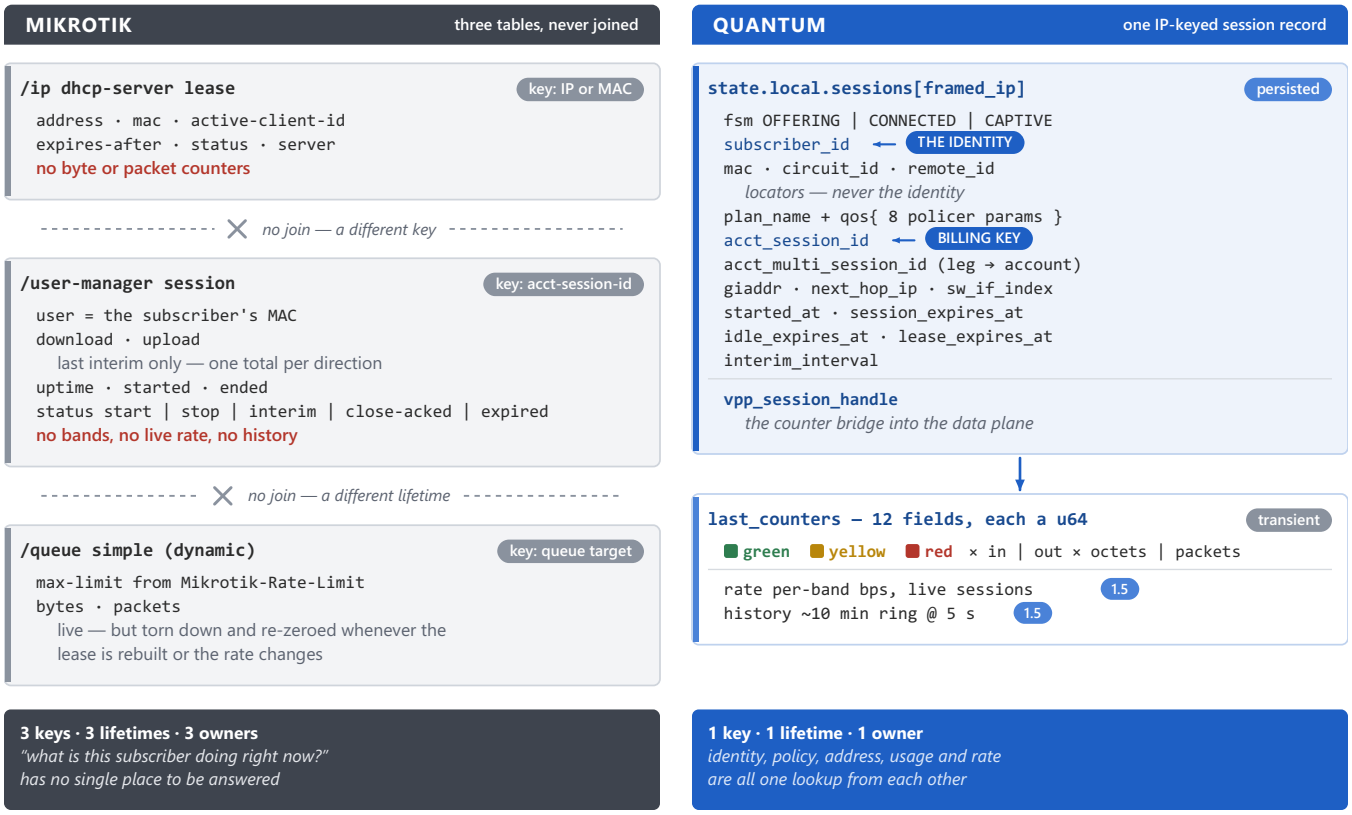
Milestone	MikroTik	Quantum CE
Install AAA	Download and install the `user-manager` package	Already present
Register the NAS	`/user-manager/router` entry per BNG, with shared secret	Not applicable — no RADIUS hop
Define a service tier	`limitation` + `profile` + `profile-limitation` (three linked objects)	One plan: eight integers
Add a subscriber	`/user-manager/user add …` per subscriber	One record, or 10,000 in one atomic call
Admit an unknown line	Not modelled — hotspot is the documented self-serve path	`unknown: captive` — walled garden, self-serve
See it working	CLI / Winbox	Console with live session view

Round 1 verdict. MikroTik arrives as a router an operator turns into a BNG: the AAA server is a separate download, and the queue topology, Option 82 strategy, accounting consumer and provisioning path are all the operator's to design. Quantum arrives as a BNG, with the glue already written.

Round 2 — The Session Table

This is the round that decides the other eight.

Where the subscriber lives



*Where a subscriber lives in each product. MikroTik needs three lookups in three tables on three different keys; Quantum needs one. Pills marked **1.5** are CE 1.5 additions — everything else in the picture ships in CE 1.4.*

MikroTik: three tables that never join

A MikroTik IPoE subscriber has no single representation. They exist as three fragments:

The DHCP lease (`/ip dhcp-server lease`), keyed by address or MAC, holds the address, the client identifier and the expiry. It holds **no byte or packet counters at all.** It is an address-allocation record, and nothing more.

The User Manager session (`/user-manager session`), keyed by `acct-session-id` , is the accounting fragment. Its complete field set is `acct-session-id` , `active` , `calling-station-id` , `download` , `ended` , `last-accounting-packet` , `nas-ip-address` , `nas-port-id` , `nas-port-type` , `started` , `status` ,

`terminate-cause`, `upload`, `uptime`, `user` and `user-address`. For usage that is **three numbers:** **`download`**, **`upload`**, **`uptime`** — and they are only as fresh as the last Accounting-Request interim that arrived over UDP.

The dynamic simple queue is the enforcement fragment. It holds the rate limits and it does have live byte and packet counters — but it is keyed by queue target, it is not part of either table above, and it is torn down and re-zeroed whenever the lease is rebuilt or the rate changes.

Three keys. Three lifetimes. Three owners. Nothing joins them, and the consequences are concrete:

- **No live per-subscriber rate exists as data.** The queue knows the bytes; the session knows the identity; nothing knows both.
- **Usage resolution is the interim interval.** Between interims, usage in the session table is stale by design.
- **A rate change destroys the counters.** Re-rating a subscriber rebuilds the queue, so the live byte counts restart from zero — mid-cycle.
- **The `expired` session status exists because the model leaks.** User Manager needs a state for "*this session stopped reporting and never sent a Stop.*" That state is the accounting-side admission that a DHCP lease gives the AAA server no reliable liveness signal.

Quantum: one record, and it is the join

A Quantum session is a first-class object, keyed by `framed_ip`, persisted across controller restarts, and carrying everything about that subscriber at once:

Field	What it is	Why it is on the session
`fsm`	`OFFERING` | `CONNECTED` | `CAPTIVE`	Service state is data, not an inference from a lease timer
`subscriber_id`	**The identity** — operator-assigned, opaque	Sessions roll up to an account
`mac` · `circuit_id` · `remote_id`	**Locators** — device and line	How the record was found; never the identity
`plan_name` + `qos{…}`	What was sold, and the eight policer parameters programmed from it	Policy and intent sit together
`acct_session_id`	UUID, node-prefixed **[1.5]**	The billing idempotency key
`acct_multi_session_id`	Subscription correlation	A CPE swap is a new leg of the same account
`framed_ip` · `giaddr` · `next_hop_ip` · `sw_if_index`	Address and attachment	Forwarding context
`vpp_session_handle`	Handle into the data plane	**The counter bridge** — Round 3
`started_at` · `session_expires_at` · `idle_expires_at` · `lease_expires_at`	Deadlines as absolute timestamps	A restarted controller rebuilds every timer from the record
`interim_interval`	Accounting cadence, **per subscriber**	Heavy accounts sampled finely; the base sampled cheaply

Alongside it, deliberately off the persisted blob so that writing counters never writes state, sits the transient half: `last_counters` — **twelve u64 counters, green/yellow/red × in/out × octets/packets** — plus `idle_streak_ms`, and **[1.5]** the sampler's per-band `rate` and short `history` ring.

That single record is the join MikroTik does not have. Identity, entitlement, address, policer state, usage and live rate are all one lookup from each other.

What the session table makes possible

The practical argument is a chain. Each property of the session table unlocks a capability, and each capability has a commercial consequence.

Session-table property	Capability it unlocks	What it is worth
One persisted record keyed by `framed_ip`	The session stays addressable for its whole life	Disconnect, re-rate or gate a subscriber at any moment — no short leases, no DORA
`subscriber_id` on the session	Sessions roll up to an account	Multi-device accounts; per-subscriber concurrency caps; a CPE swap keeps the customer
`plan_name` + `qos{8}` together	What was sold sits beside what was programmed	A tier change is a one-field write, not queue archaeology
`vpp_session_handle` on the record	Direct access to that subscriber's data-plane counters	Throughput readable at any cadence, with no RADIUS round trip
12 band counters per session	Usage split by conformance colour	Conformance billing, burst billing and upsell signals — Round 4
u64 end to end	No 32-bit wrap, no gigawords arithmetic	Multi-terabyte subscribers bill correctly
`acct_session_id` as idempotency key	At-least-once delivery is safe	Billing can retry a failed drain without double-counting
`boot_epoch` on every record	The reboot generation is visible	Sessions a restart orphaned are sealed, not silently lost
Per-session `interim_interval`	Accounting cadence is a per-subscriber decision	Sample a business account at 60 s and the residential base at 600 s
Five-state FSM including `CAPTIVE`	Non-payment is a **state**, not a disconnect	Dunning and self-serve top-up without losing the session
Absolute deadlines on the record	Timers are data, not process state	A controller restart rebuilds every deadline exactly

Round 2 verdict. MikroTik's three-table split is not an implementation detail that a release could tidy up — the lease, the accounting session and the queue have genuinely different lifetimes in that architecture. Quantum's single session record is the reason the next three rounds go the way they do.

Round 3 — Throughput, Subscriber by Subscriber

Round 2 established *where* usage lives. This round is about *what is measured*, and how good the measurement is.

Two outcomes versus six

MikroTik's enforcement is a token-bucket simple queue. A packet is **passed** or **dropped**. Those are the only two outcomes the mechanism can distinguish, so they are the only two outcomes it can count — and only the passed count reaches the accounting record.

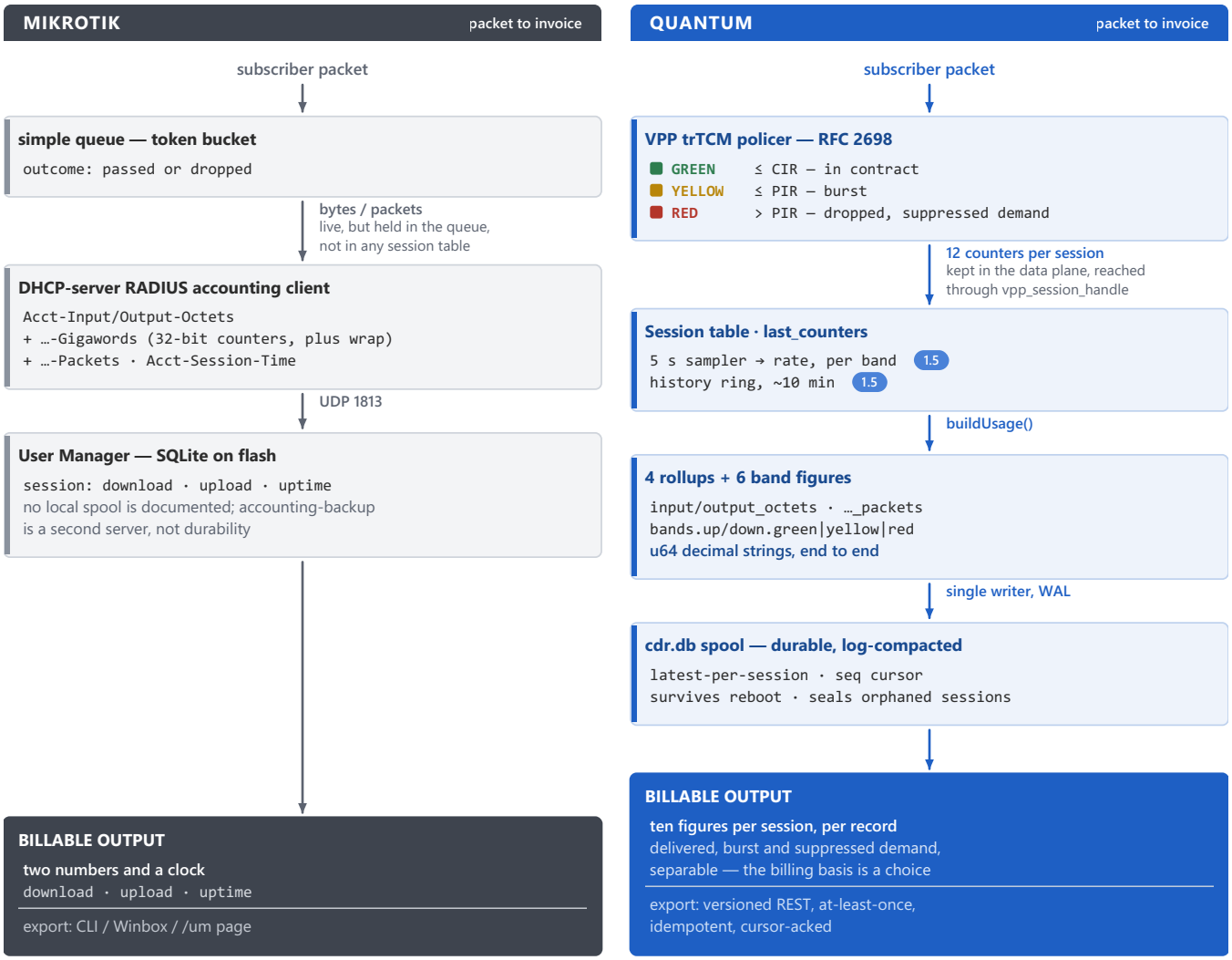
Quantum's enforcement is an **RFC 2698 two-rate three-colour marker** in the VPP data plane. Every packet is coloured:

Colour	Meaning	Commercial reading
Green	Within the committed rate (CIR)	In-contract traffic — what the customer paid for
Yellow	Above CIR, within the peak rate (PIR)	Burst — delivered, but out of contract
Red	Above PIR	Dropped — **demand the plan could not satisfy**

Three colours, two directions, octets and packets: **twelve counters, per subscriber, maintained in the data plane**, reachable through the `vpp_session_handle` on the session record.

Red is the one no competitor at this price point has. It is not an error counter — it is *suppressed demand*, measured per subscriber. A subscriber with a persistent non-zero red count is asking for a faster plan.

The measurement chain, end to end



*From policed packet to billable record. Both chains end in a billing export; only one of them carries the shape of the traffic with it. The **1.5** pills mark the server-side telemetry additions; the twelve band counters, the* `buildUsage` *rollups, the per-band billing record and the durable spool all ship in CE 1.4.*

Reading it at any cadence

Because the counters live in the data plane and the session record holds a handle to them, reading throughput costs no RADIUS exchange and no state write. Quantum exposes three distinct reads, each tuned to a different question:

Read	Question it answers	Cost
`GET /bng/counters`	"What is the whole box doing, by band?"	Aggregate — no per-session fan-out. Safe to poll at any rate
`GET /bng/counters/{ip}`	"What is this subscriber doing right now?"	One session, read straight from the data plane
`GET /bng/sessions/{ip}`	"Show me this subscriber's drawer"	Session detail + `last_counters` + per-band `rate` + `history` **[1.5]**
`GET /bng/sessions/top` **[1.5]**	"Who are my top talkers?"	Ranked over **every** session, served from a sampler already running

The sampler **[1.5]** is worth a note, because it shows the architectural dividend. On a five-second clock it reads every live session's counters in bulk cohorts, differences them to bits per second, and maintains an instantaneous per-band rate for *every* session plus a short history ring for the working set. It never touches the write path. A top-talkers table over the entire subscriber base is therefore a cheap synchronous read — because the session table already had somewhere to put the answer.

On MikroTik the same question has no cheap form. The rate is in the queues, the identity is in the session table, and joining them is work the operator does outside the product.

Precision

	MikroTik	Quantum
On-the-wire counter width	`Acct-Input/Output-Octets` are 32-bit, extended by `…-Gigawords`	**u64**
Representation in transit	Integer attributes requiring gigawords reassembly	**u64 decimal strings**, end to end
Failure mode at volume	A reassembly error or a missed gigawords attribute mis-bills by 4 GB multiples	None — no reassembly step exists
Packets	Reported on interim	Per band, both directions

Quantum carries counters as decimal strings deliberately: a 64-bit octet count exceeds the exact-integer range of a double, so anything that passes through a JSON number risks silent rounding at scale. The string representation survives the controller, the REST layer, the spool and the management plane without a single lossy conversion. On a 2 TB-per-month subscriber that is the difference between a bill and an argument.

Round 3 verdict. MikroTik measures what a token bucket can tell it: bytes passed, as of the last interim. Quantum measures what the subscriber was offered, what was delivered in contract, what was delivered as burst, and what had to be dropped — per subscriber, per direction, continuously.

Round 4 — The Billing Data

This round is the payoff. The session table and the band counters exist so that the embedded AAA can produce a billing record an ISP can actually run a business on.

What each product hands the billing system

	MikroTik User Manager	Quantum local AAA
Record granularity	One session row, updated by interim	One CDR per session, `start` | `interim` | `stop`
Usage fields	`download`, `upload`, `uptime`	4 rollups + **6 band figures**, octets **and** packets
Conformance detail	None	Green / yellow / red, per direction
Identity on the record	`user` = the MAC	`subscriber_id`, plus `acct_multi_session_id`
Line context	`nas-port-id`, `calling-station-id`	`circuit_id`, `remote_id`, `mac`, `framed_ip`, `plan_name`
Counter width	32-bit + gigawords	u64 decimal strings
Export mechanism	CLI / Winbox / `/um` page	`GET /bng/accounting/records?since=` → `POST …/ack`
Delivery guarantee	UDP accounting; no documented local spool	**At-least-once, idempotent on** `acct_session_id`
Durability	SQLite on device flash	Durable spool, single-writer, WAL, off-thread
Consumer outage	Not documented	Spool retains; cursor resumes; backpressure watermarks at 50k / 25k
Unclean restart	`expired` status; reconciliation not documented	**Reboot-sealed** `stop` with `terminate_cause: NAS_REBOOT`

The Quantum record, field by field

Field	Note
`acct_session_id`	The idempotency key — a retried drain can never double-count
`subscriber_id`	The account key
`acct_multi_session_id`	Correlates legs of one subscription across device changes
`circuit_id` · `remote_id` · `mac` · `framed_ip` · `plan_name`	Line, device, address and policy context on every record
`record_type`	`start` | `interim` | `stop`
`input_octets` · `output_octets` · `input_packets` · `output_packets`	Materialised band sums, as u64 decimal strings
`usage.bands.{up,down}.{green,yellow,red}`	**Bytes and packets for all six cells**
`session_time_s` · `session_start_ts` · `record_ts` · `boot_epoch`	Timing and reboot generation
`terminate_cause`	On `stop` — RFC 2866 numerics
`class_attr`	base64; reserved for RADIUS `Class` round-trip under the optional connector

A note for whoever writes the billing integration. The four rollups are the sum of all three bands, so `output_octets` is traffic **offered** to the subscriber. Delivered traffic is green + yellow. Both readings are available from the same record, and choosing between them is a deliberate commercial decision rather than an artefact of the measurement. No other product in this class gives the operator that choice, because no other product keeps the bands.

Worked example — one subscriber, one month

An illustrative residential subscriber on a 100 Mbps committed / 150 Mbps peak plan, over a 30-day cycle. The figures are illustrative; the *fields* are exactly what the two products produce.

What Quantum's CDR contains:

Direction	Green (in contract)	Yellow (burst)	Red (dropped)	Offered total
Downstream	742.31 GB	38.65 GB	9.10 GB	790.06 GB
Upstream	61.20 GB	2.88 GB	0.61 GB	64.69 GB

What MikroTik's session row contains:

Field	Value
`download`	≈ 780.96 GB
`upload`	≈ 64.08 GB
`uptime`	29 d 14 h 22 m

Both are true. Only one of them can be worked with:

Commercial question	MikroTik	Quantum
Flat-rate monthly volume	**Yes**	**Yes**
Bill in-contract traffic only	No	**Yes** — green
Charge separately for burst above CIR	No	**Yes** — yellow
Report SLA conformance (% in contract)	No	**Yes** — green ÷ (green + yellow) = 95.0 % down
Identify subscribers whose demand exceeds their plan	No	**Yes** — 9.10 GB of red is an upsell signal
Apply a fair-use policy to in-contract traffic only	No	**Yes**
Settle a wholesale bill on delivered vs offered	No	**Yes** — both are in the record
Defend a disputed bill with packet counts as well as bytes	Partial	**Yes** — packets, per band, per direction

The 9.10 GB of red is the line that matters commercially. On MikroTik that traffic is invisible: it was dropped, so it was never counted, so nothing in the system records that the customer asked for more than they bought. On Quantum it is a number attached to a `subscriber_id` — and a retention and upsell programme can be built on it.

When the billing consumer goes away

Accounting is only worth as much as its weakest hour.

MikroTik. Accounting is UDP to the User Manager. `accounting-backup` on `/radius` offers a second server; no local spooling of unsent accounting records is documented. The `expired` session status implies that sessions which stop reporting are reaped rather than reconciled.

Quantum. The CDR spool is the system of record for usage until the consumer acknowledges it:

- A single dedicated writer thread owns the SQLite spool with WAL — the checkpoint never lands on the event loop that processes subscriber onboarding.
- The spool is **log-compacted, latest-per-session**: a session updated fifty times since the cursor appears once, at its latest. Spool growth tracks the subscriber count, not the elapsed outage.
- Drain is a cursor contract — `?since=<seq>` then `POST …/ack` — at-least-once and idempotent on `acct_session_id`. Consumer lag is an observable number: `high_water − last acked seq`.
- An ack prunes terminated sessions at or below the cursor and **retains active ones**, so they remain available for recovery.
- Backpressure watermarks at 50,000 / 25,000 records, and a worker that crashes is respawned with backoff.
- After an unclean restart, any session left open under a prior `boot_epoch` and not seed-and-continued is **sealed with a `stop` carrying `NAS_REBOOT`** and its last-known usage. The ledger never carries a session that a reboot silently ended.

That last point is the direct answer to MikroTik's `expired`. Both products know that a subscriber session can end without anyone being told. One of them has a status for it; the other closes the record and bills it.

Round 4 verdict. MikroTik produces two numbers and a clock, exported by hand, over a protocol with no durability story. Quantum produces a ten-figure, conformance-resolved, u64-precise record per session, drained over a documented idempotent cursor from a spool engineered to survive both the consumer and the box.

Round 5 — Changing a Live Subscriber

An operator's day is made of mid-session changes: a tier upgrade, a payment that clears, a payment that does not, an abuse complaint. This round is about whether the product can act on a subscriber who is already online.

MikroTik on IPoE: the lease is not a session

MikroTik supports Disconnect-Message and CoA, and they work on **PPPoE**, where a control-plane session with a keepalive exists. On the **DHCP/IPoE** path the field record across many years and versions is consistent: a Disconnect-Request against a lease comes back as `Session-Context-Not-Found`, or as a `Disconnect-NAK` with `Unsupported-Extension`. The settled community workaround is **short lease times of one to five minutes**, so that policy changes take effect at the next renewal.

MikroTik is also explicit in its own documentation that **IP addresses, pools and routes cannot be changed by CoA** — the session must be disconnected — and that **Packet of Disconnect is not supported**.

There is a further structural point. User Manager's attribute engine targets only `access-accept` and `access-challenge`; CoA-Request is not an available target, and there is no AAA API. So even where Dynamic Authorization works, User Manager can only emit what its own internal quota evaluator decides to emit. An external billing system cannot say *"re-rate subscriber X now."* It can rewrite the profile and wait for the next authentication event.

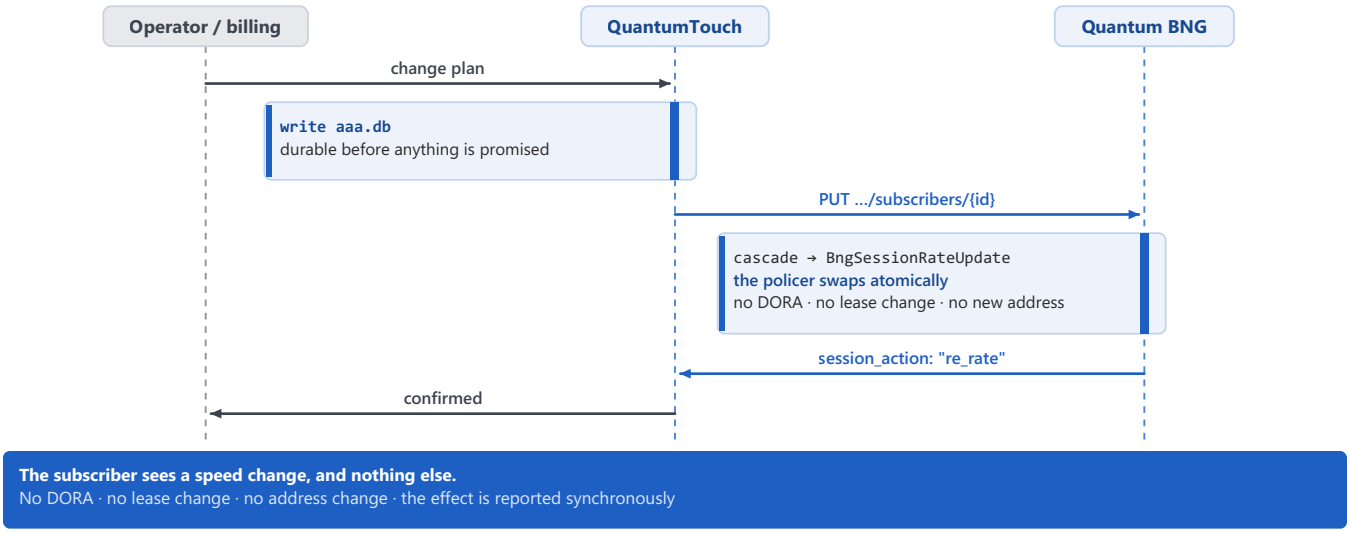
The defence usually offered — *"there is no mechanism in DHCP to revoke a lease and tell the client"* — is true about the **client** and false about the **BNG**. A BNG does not need the client's cooperation to change policy. It owns the forwarding state, the policer, the filter and the FIB entry for that subscriber's address. Re-rating is a data-plane action, not a DHCP action.

Quantum: one record write, live effect, reported back

On Quantum the operator edits the subscriber record. The write cascades into the live session and the API reports what happened:

Cascade	Trigger	Effect on the live session
`re_rate`	Plan changed	`BngSessionRateUpdate` — the policer swaps atomically. No DORA, no address change
`revoked`	Status → suspended	Session moves to `CAPTIVE` — walled garden, with configurable extra garden ports
`disconnected`	Record deleted	Session torn down, `stop` CDR written
`activated`	Suspended → active	Captive session restored to its plan rates
`deferred_dora`	Change needs a fresh handshake	Applied on the next DORA, reported as such
`none`	Nothing live to change	Record updated

LIVE RE-RATE



A plan change applied to a live session. QuantumTouch makes the change durable in `aaa.db` *before it is promised to anyone, and the BNG reports what it actually did in the same response.*

Two details make this operationally different rather than merely faster.

The effect is reported synchronously. `session_action` comes back in the API response, so a billing system knows the change landed rather than hoping it did. In a bulk operation every row carries its own action.

Non-payment is a state, not a disconnect. `CAPTIVE` keeps the session, the address and the accounting alive while restricting reachability to the walled garden. The subscriber can self-serve a top-up and be restored with `activated` — no re-onboarding, no lost session continuity, and a continuous billing record across the whole episode. The alternative model, which MikroTik's quota engine implements, is to disconnect on quota exhaustion.

Data caps. Quantum carries an enforcement overlay — `throttle` to a floor tier, `captive` to a top-up page, or `clear` — applied both at admission and to live sessions, with an `enforcement/bulk` call for billing-cycle mass restore.

Round 5 verdict. On IPoE, MikroTik's answer to "change this subscriber now" is to shorten the lease and wait. Quantum's is one record write with the result in the response.

Round 6 — Who the Subscriber Is

A quieter round, but it is the one that decides whether the product can model a customer.

MikroTik: the MAC is the identity

On the DHCP path RouterOS sends an Access-Request in which `User-Name` **is the client's MAC address** and `User-Password` is empty. The User Manager session table's `user` column therefore holds a MAC. Identity and locator are the same object, and three consequences follow directly:

- **A CPE swap is a new customer.** New MAC, new user, new session history.
- **Two devices are two unrelated subscribers.** There is no account to which both belong.
- **Usage history is attached to hardware, not to a person.** Replace the router, lose the record.

MAC-based authorisation is standard IPoE practice; carrying the MAC as the *identity* is the part that does not scale to a subscriber base.

Quantum: identity, and the locators that find it

Quantum's local AAA provider is explicit on this point in its own source: the BNG never creates a subscriber's identity. The data plane presents **locators** — circuit-id, MAC — and the record store returns a stable `subscriber_id`.

- **MAC** is the session / device key. Never an identity.
- **circuit-id** is the attachment / line locator.
- `subscriber_id` is the identity, operator-assigned and opaque.

A record carries a `match[]` list, so one subscriber may be reachable by several locators — a line and two devices, or a line that keeps its account through a CPE replacement. The resolver takes locators in and returns an identity plus an outcome: `accept`, `captive` or `reject`, with the enforcement overlay applied.

	MikroTik	Quantum
Identity	The MAC	Opaque `subscriber_id`
Locators	Identical to the identity	`match[]` — many locators, one subscriber
CPE swap	New subscriber	Same subscriber, new leg (`acct_multi_session_id`)
Multi-device account	Not modelled	Native
Unknown line	Not modelled on the DHCP path	`unknown: captive` — self-serve onboarding
Static IP on IPoE	`Framed-IP-Address` from RADIUS	`framed_ip` on the record

Round 6 verdict. MikroTik identifies hardware. Quantum identifies customers, and treats hardware as one of several ways to find them.

Round 7 — Provisioning, and What Survives an Upgrade

Provisioning

	MikroTik	Quantum CE 1.4
Interface	CLI, Winbox, WebFig	Versioned REST, role-based keys, audit log
AAA API	**None documented**	13 routes under `/bng/aaa`
One subscriber	One CLI invocation	`PUT …/subscribers/{id}`
Whole base	A script per subscriber	**Atomic bulk upsert to 10,000**, with per-row `session_action`
Mass delete	Per subscriber	`…/bulk_delete`
Mass enforcement restore	Not modelled	`…/enforcement/bulk`
Service tier definition	Three linked objects (`limitation` + `profile` + `profile-limitation`)	One plan: eight integers
Config validation	—	`config_ok` false on a dangling plan reference → health `degraded`

The question nobody asks until it bites

Where does the subscriber database live, and what happens when you upgrade the box?

On MikroTik the answer is: on the device's flash, in User Manager's SQLite file, and that is the only copy. No replication. No HA. No external database backend. No external identity source — no LDAP, no AD, no SQL lookup.

Quantum answers it differently, and the reason is instructive. The controller package's installation script clears its own configuration database on every install and upgrade. An AAA service that kept 5,000 subscriber accounts there would lose them on a routine package update. So the architecture was changed rather than the symptom patched: **QuantumTouch is the system of record for AAA.**

Layer	Role
QuantumTouch `aaa.db`	**System of record.** Subscribers, enforcement, sync state, and accounts it did not create. Survives controller reinstall
Quantum BNG controller	**Execution layer.** Holds the running configuration and applies it to the data plane
Reconciler, 60 s	Converges the two: diff and push the full record, detect a wipe, drain accounting

Three rules govern it, and each one exists because the alternative was tried:

- **Durable first, then push.** The irreversible half of an operation never happens before the durable half.
- **Never delete a subscriber we did not create.** Accounts found upstream that QuantumTouch did not provision are recorded as `unmanaged` and are never auto-removed; the console offers Adopt or Remove.
- **A push failure is not an operator error.** The write succeeded durably; the reconciler keeps trying and the console says so.

Wipe detection uses a two-pass debounce before bulk re-provisioning, so a single slow poll cannot trigger a mass rewrite.

Round 7 verdict. MikroTik keeps the subscriber base in one file on the device that forwards the traffic. Quantum keeps it deliberately outside the BNG, and treats the BNG as a cache that can be rebuilt — which is the only architecture under which "upgrade the BNG" is a routine act.

Round 8 — Scale, Licensing and Operating Cost

	MikroTik	Quantum
What caps sessions	RouterOS licence level	Quantum capacity tier
Free tier	Low licence levels permit a small number of concurrent sessions	**1,000 sessions, unregistered**
Free registered tier	—	**5,000 sessions** — free self-service Registration Unlock, perpetual, one per BNG, unlimited BNGs
Commercial tiers	Higher licence levels	10k / 25k / 50k / 100k, applied immediately on activation
Cap visibility	An error string when the cap is hit	`GET /bng/health` reports `max`, `max_source`, `at_cap`; `GET /quantum/license/status` returns the verified grant map
Licence integrity	Licence key on device	TPM-sealed, hash-checked cache; expired or tampered reverts to no grants

Two differences matter in a procurement conversation.

The free tier is a production tier. 1,000 subscribers without any registration, and 5,000 after a free self-service unlock, is a real small-ISP deployment — not an evaluation allowance. An operator can build a business on Community Edition and upgrade when the subscriber base, not the licence, demands it.

The cap is reported with its provenance. A management plane can render the tier honestly because the API says both the number and where it came from. On MikroTik, discovering the ceiling is an operational event.

Round 8 verdict. Quantum's free tier is five times the size of a plausible small ISP's base, and its paid tiers go an order of magnitude beyond what a RouterOS licence addresses.

Round 9 — Where MikroTik Wins

A bake-off that scores nine-nil is not a bake-off, it is a brochure. MikroTik wins this round, and the wins are genuine.

MikroTik capability	Detail	Quantum position
EAP and 802.1X	Full outer and inner method suite: EAP-TLS, EAP-TTLS, PEAP, EAP-MSCHAPv2, with per-group method constraints	Not applicable to IPoE scope
TOTP two-factor	Per-user `otp-secret`	Not offered
RadSec	TLS-protected RADIUS on TCP 2083, selectable **per NAS**, with its own certificate	Roadmap
Arbitrary vendor attributes	`/user-manager/attribute` defines any attribute under any IANA PEN, with six value types	Roadmap, with the optional connector
Calendar-tiered prepaid	`from-time` / `till-time` and `weekdays` on the profile-limitation join; `validity`, `price`, PayPal self-purchase	**A genuine product gap** — Quantum has plans and caps, but no time-of-day or weekday tiers
Multi-service AAA	One server for PPPoE, hotspot, wireless, IPoE and router login	Quantum's AAA is IPoE-scoped by design
Price	The package is free with RouterOS	Community Edition is also free, at larger scale

Three of these — EAP, TOTP, RadSec — are outside the IPoE path by design: an IPoE subscriber authorises with a MAC and a line identifier, and no amount of EAP machinery changes that. On the DHCP path MikroTik's own entire authentication subsystem is bypassed too.

Calendar-tiered prepaid is the one to take seriously. For a prepaid or hotspot-adjacent operator, off-peak tiers and timed vouchers are a real product requirement, and MikroTik has them today. It belongs on Quantum's roadmap, not in a rebuttal.

RadSec is the one to watch. A TLS-protected RADIUS transport, selectable per NAS, is a serious capability at this price point, and it is the right answer for an operator whose RADIUS traffic crosses untrusted transport.

Round 9 verdict. MikroTik is the broader AAA server, and on calendar-tiered prepaid it has a capability Quantum does not have at any price. Quantum's AAA is built for IPoE subscribers only, which is why most of this table sits outside its scope by design — the calendar tiers do not, and they belong on the roadmap rather than in a rebuttal.

The Scorecard

Session model and identity

Capability	MikroTik	Quantum CE 1.4
Subscriber representation	**Three tables, three keys, three lifetimes**	**One persisted, IP-keyed session record**
Identity model	The MAC *is* the identity	Opaque `subscriber_id`; locators match to it
Multi-device / multi-locator account	Not modelled	`match[]`
CPE swap continuity	New subscriber	Same account, new leg
Service state	Inferred from a lease	Five-state FSM including `CAPTIVE`
Unknown-line admission	Not modelled on DHCP	`unknown: captive` self-serve
Static IP on IPoE	`Framed-IP-Address`	`framed_ip` on the record

Measurement and billing

Capability	MikroTik	Quantum CE 1.4
Per-subscriber live rate	No single home for it	Per-band bps, every session **[1.5]**
Short rate history	—	~10 min ring @ 5 s **[1.5]**
Top-talkers over the base	—	Ranked read over all sessions **[1.5]**
Usage granularity	`download` · `upload` · `uptime`	**Octets and packets per QoS band**, both directions
Conformance detail	None	Green / yellow / red
Suppressed demand (dropped)	Not reported	**Red counters, per subscriber**
Counter precision	32-bit + gigawords	**u64 decimal strings end to end**
Billing export	CLI / Winbox / `/um`	Versioned REST, cursor + ack
Delivery guarantee	UDP accounting	At-least-once, idempotent on `acct_session_id`
Accounting durability	No documented local spool	Durable off-thread spool, WAL, single writer, backpressure
Consumer-outage behaviour	Not documented	Retained, log-compacted, lag observable
Unclean-restart reconciliation	`expired` status	**Reboot-sealed** `stop` , `NAS_REBOOT`

Control of a live subscriber

Capability	MikroTik	Quantum CE 1.4
Mid-session re-rate (IPoE)	**No** — field record	**Yes** — `re_rate` → `BngSessionRateUpdate`
Mid-session disconnect (IPoE)	**No** — `Session-Context-Not-Found`	**Yes**
Change address / pool / route by CoA	**No** — stated by MikroTik	Not applicable — no CoA hop
Packet of Disconnect	**Not supported** — stated by MikroTik	Not applicable
Policy push from OSS / BSS	No trigger surface; next auth only	REST record write, effect reported synchronously
Gate to walled garden	Hotspot only	`revoked` → `CAPTIVE`, configurable garden ports
Data caps / fair use	Quota → disconnect	Overlay: `throttle` or `captive`, at admission **and** live
Billing-cycle mass restore	Not modelled	`enforcement/bulk`
Concurrency control	`shared-users` (silent disconnect)	Subscriber-scoped session view + explicit disconnect

Platform and operations

Capability	MikroTik	Quantum CE 1.4
Completeness at download	AAA is a separate package; glue is yours	**Everything in the box; traffic in under an hour**
AAA management UI	**None in v7**	QuantumTouch console
Provisioning API	None documented for AAA	Versioned REST, role-based keys, audit log
Bulk provisioning	One per CLI call	Atomic to **10,000**
Subscriber DB location	SQLite on the forwarding device — only copy	**QuantumTouch is the system of record**
Survives a BNG upgrade	The file is the only copy	Reconciler re-provisions from `aaa.db`
HA / replication	None	System of record is external to the BNG
External identity source	None	Roadmap (optional connector)
Config validation	—	`config_ok` → health `degraded`
Free-tier scale	Small, licence-dependent	**1,000 / 5,000 registered**
Commercial scale	Licence levels	To **100,000**

Breadth outside IPoE

Capability	MikroTik	Quantum CE 1.4
EAP / 802.1X suite	**Yes**	No — IPoE scope
TOTP two-factor	**Yes**	No
RadSec	**Yes**, per NAS	Roadmap
Arbitrary vendor attributes	**Yes**, any PEN	Roadmap
Calendar-tiered prepaid	**Yes**	**Gap**
One server for PPPoE / hotspot / wireless	**Yes**	No — IPoE scope

How to Decide

Three questions settle it in a customer conversation.

1. "Is IPoE the business, or one service among several?" If the box has to be a RADIUS server for PPPoE, hotspot, wireless and router login as well, MikroTik's breadth is the right answer and we should say so. If the business is a residential or business IPoE subscriber base, breadth outside IPoE buys nothing and the session model decides everything.

2. "Do you bill on usage, or on a flat rate?" A flat-rate operator who never re-rates mid-cycle and never disputes a bill can live with two numbers and a clock. Everyone else needs conformance-resolved, u64-precise, durably spooled records tied to a subscriber identity — and that is Rounds 3 and 4.

3. "When a customer calls to upgrade, what happens?" On MikroTik: change the profile and wait for the lease to renew — which is why the deployments run one-to-five-minute leases. On Quantum: one record write, the policer swaps atomically, the API confirms `re_rate`, and the subscriber sees a speed change and nothing else.

The one-paragraph version

MikroTik gives an ISP a capable multi-service RADIUS server that was not built around a subscriber session, bolted to a DHCP server that was not built to be a BNG. The subscriber ends up spread across a lease, an accounting row and a queue, which is why there is no live per-subscriber rate, no conformance detail in the usage record, and no way to change a live IPoE subscriber without waiting for a lease to expire. Quantum starts from the session: one IP-keyed record holding identity, plan, address, twelve data-plane band counters and the handle that reads them — which is why the embedded AAA can emit a conformance-resolved billing record per session, why a plan change takes effect in the policer immediately, and why the whole thing is complete and carrying traffic within an hour of the download.

Appendix A — MikroTik Reference

Condensed from BlueDot's RouterOS 7 `user-manager` teardown. Retained here so the body can stay a comparison.

Packaging and lineage

Aspect	Detail
Delivery	Optional downloadable RouterOS package, `user-manager`
Architectures	All RouterOS architectures **except SMIPS**
Lineage	Complete rewrite for RouterOS v7; v5/v6 User Manager is a different product
Legacy migration	`/user-manager/database migrate-legacy-db`
Persistence	On-box **SQLite** `.db` file on device flash
Admin interface	**Removed in v7.** CLI, Winbox or WebFig only
End-user interface	`https://<router>/um/` — profile view, statistics, PayPal purchase
Licensing	Concurrent active sessions capped by RouterOS licence level

The v7 rewrite is the key lineage fact: most third-party tutorial content and most negative performance folklore refers to v5. The v7 engine added the capability that makes User Manager arguably a general-purpose RADIUS server — operator-definable RADIUS attributes with an arbitrary vendor ID.

Documented absences

For a product of this maturity, undocumented is functionally unsupported.

- No RADIUS proxy or forwarding; no realm routing, no home-server pools.
- No high availability or replication — the SQLite database is local.
- No external database backend: no MySQL, PostgreSQL or ODBC.
- No external identity source: no LDAP, AD or SQL user lookup.
- No programmatic AAA API for provisioning or on-demand CoA injection.
- No documented Accounting-On / Accounting-Off handling, and therefore no documented NAS-restart session reconciliation.

RFC posture

RFC	Subject	Position
2865	RADIUS	Stated compliance
3579	RADIUS support for EAP	Stated compliance
2866	RADIUS Accounting	Implemented in substance; not cited
2869 §5.16	`Acct-Interim-Interval`	Honoured on the PPP path
3576	Dynamic Authorization	**Cited — obsolete.** Superseded by RFC 5176
5176	Dynamic Authorization (current)	Not cited
6613 / 6614	RADIUS over TCP / RadSec	Implemented; not cited
6929	Extended attribute space	No evidence
3046	DHCP Relay Agent Information	Implemented on the DHCP path
4679 / TR-101	DSL Forum VSAs	Claimed via `support-broadband-tr101`

MikroTik explicitly states that IP addresses, pools and routes cannot be changed by CoA, and that Packet of Disconnect is not supported.

Messaging channels

Channel	Endpoint
Authentication	UDP 1812
Accounting	UDP 1813
RadSec	TCP 2083, per-NAS via `protocol` on `/user-manager/router`
Incoming CoA / DM	`/radius incoming`, `accept=yes`, port 1700 by default on the client side; `coa-port` per NAS on the server side
Client config	`/radius` — `address`, `secret`, `service=dhcp,ppp,hotspot,…`, `timeout` 1100 ms, VRF, `src-address`, `accounting-backup`, `require-message-auth`

`require-message-auth` is the BlastRADIUS (CVE-2024-3596) control and should be enabled on any new deployment.

The DHCP seam

```
/radius
  add address=10.0.0.1 secret=VERYsecret123 service=dhcp

/ip/dhcp-server
  set dhcp1 use-radius=yes
```

`use-radius`	Behaviour
`no`	Default. No RADIUS involvement
`yes`	RADIUS consulted for lease assignment and receives accounting
`accounting`	Accounting only; leases come from local pools

Access-Request for a DHCP client carries `User-Name` (the MAC), `User-Password` (empty), `NAS-Identifier`, `NAS-IP-Address`, `NAS-Port` (ifIndex), `NAS-Port-Id` (interface name), `NAS-Port-Type` (Ethernet), `Calling-Station-Id` (`active-client-id`), `Called-Station-Id` (server instance) and `Framed-IP-Address`, plus `Mikrotik-Host-IP` (VSA 10).

Access-Accept honoured on the DHCP path — six documented attributes:

Attribute	Effect
`Framed-IP-Address`	Assign this specific IP
`Framed-Pool`	Allocate from this named pool
`Session-Timeout`	**Becomes the DHCP lease time**
`Mikrotik-Rate-Limit` (VSA 8)	Creates a dynamic simple queue
`Ascend-Data-Rate`	Alternative rate limit, lower precedence
`Ascend-Xmit-Rate`	Alternative TX rate limit

Precedence is `Mikrotik-Rate-Limit` > Ascend > WISPr. Per-lease RADIUS rate limiting for DHCPv4 arrived in RouterOS 6.43. The PPP surface additionally honours `Filter-Id`, `Framed-Route`, `Framed-IP-Netmask`, `Mikrotik-Mark-Id`, `Mikrotik-Address-List`, `Mikrotik-Recv-Limit` / `Xmit-Limit`, `Mikrotik-Total-Limit`, `Idle-Timeout`, `Port-Limit`, `Acct-Interim-Interval`, `Class`, `Mikrotik-Group` and the IPv6 set — a substantially larger documented table.

Accounting on the DHCP path. Interim updates are mandatory for the quota engine; without them quota exhaustion cannot be detected and no disconnect is issued.

Message	Adds
Start	Access-Request attributes + `Acct-Status-Type=Start`, `Acct-Authentic`
Interim	+ `Acct-Session-Time`, `Acct-Input/Output-Octets`, `…-Gigawords`, `…-Packets`, `Acct-Delay-Time`
Stop	+ `Acct-Terminate-Cause`

Option 82 and TR-101. `add-relay-info` defaults to `no`; when enabled, circuit-ID is the ingress interface MAC and remote-ID is the client MAC — no line identity. `support-broadband-tr101` adds suboptions 0x81 / 0x82 (actual upstream / downstream rate), documented as included in RADIUS Access-Request and Accounting-Request messages.

Vendor attribute surface — PEN 14988

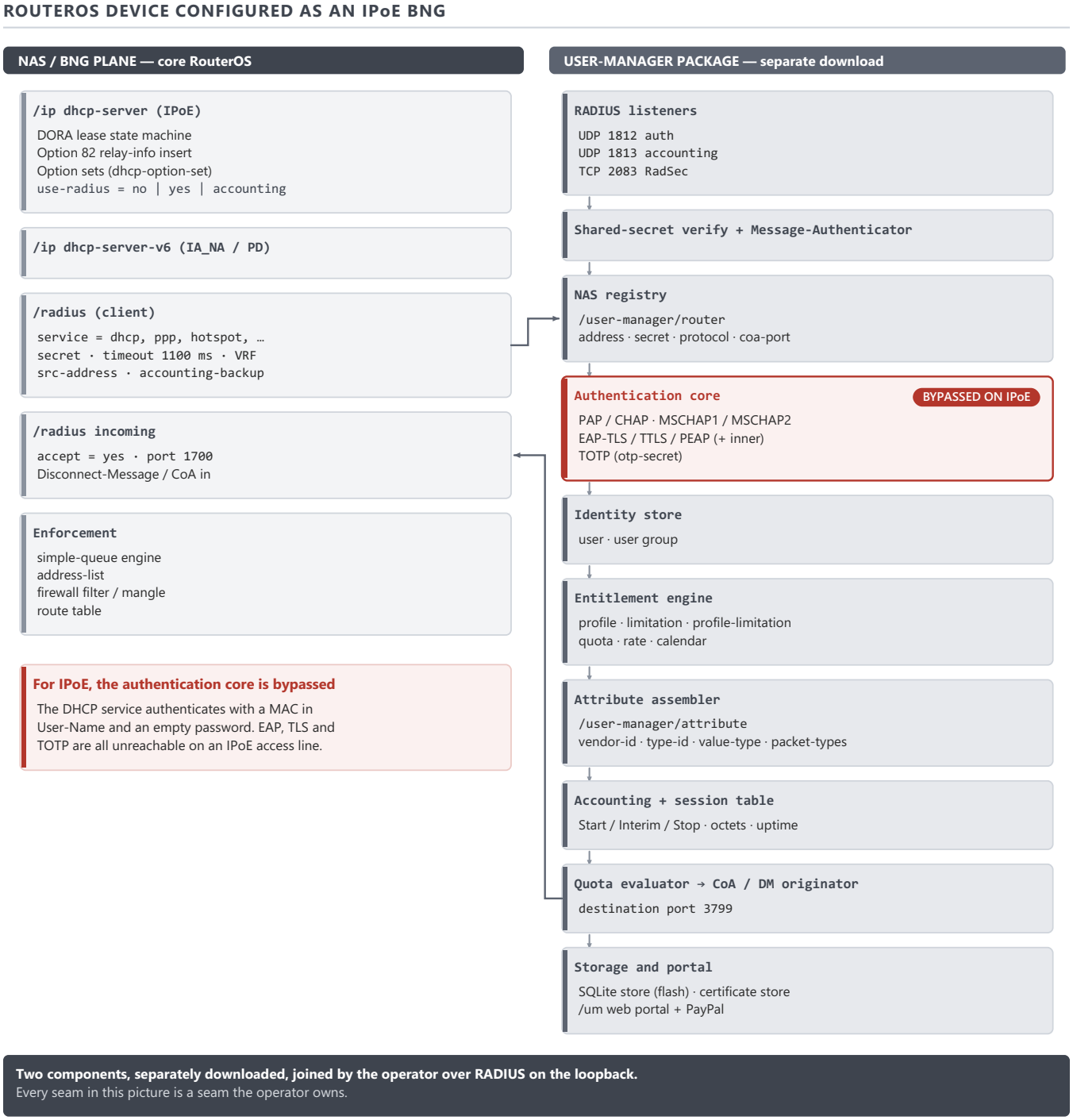
ID	Attribute	Applies to
1	`Mikrotik-Recv-Limit`	PPP
2	`Mikrotik-Xmit-Limit`	PPP
3	`Mikrotik-Group`	PPP, login
4	`Mikrotik-Wireless-Forward`	Wireless
5	`Mikrotik-Wireless-Skip-Dot1x`	Wireless
6	`Mikrotik-Wireless-Enc-Algo`	Wireless
7	`Mikrotik-Wireless-Enc-Key`	Wireless
8	**`Mikrotik-Rate-Limit`**	**PPP, DHCP, hotspot**
9	`Mikrotik-Realm`	PPP
10	**`Mikrotik-Host-IP`**	**DHCP (Access-Request)**
11	`Mikrotik-Mark-Id`	PPP
12	`Mikrotik-Advertise-URL`	Hotspot
13	`Mikrotik-Advertise-Interval`	Hotspot
14	`Mikrotik-Recv-Limit-Gigawords`	PPP
15	`Mikrotik-Xmit-Limit-Gigawords`	PPP
16	`Mikrotik-Wireless-PSK`	Wireless
17	`Mikrotik-Total-Limit`	PPP
18	`Mikrotik-Total-Limit-Gigawords`	PPP
19	`Mikrotik-Address-List`	PPP
20	`Mikrotik-Wireless-MPKey`	Wireless
21	`Mikrotik-Wireless-Comment`	Wireless
22	`Mikrotik-Delegated-IPv6-Pool`	PPP, DHCPv6

ID	Attribute	Applies to
23	`Mikrotik-DHCP-Option-Set`	**DHCP**
24	`Mikrotik-DHCP-Option-Param-Str1`	**DHCP**
25	`Mikrotik-DHCP-Option-Param-Str2`	**DHCP**
26	`Mikrotik-Wireless-VLANID`	Wireless
27	`Mikrotik-Wireless-VLANID-Type`	Wireless
28	`Mikrotik-Wireless-Minsignal`	Wireless
29	`Mikrotik-Wireless-Maxsignal`	Wireless
30	`Mikrotik-Switching-Filter`	Switching

Other vendor dictionaries recognised: Microsoft (PEN 311), Ascend (529), WISPr (14122), DSL Forum (3561).

Changeable by CoA (PPP context): `Mikrotik-Rate-Limit`, `Mikrotik-Mark-Id`, `Mikrotik-Address-List`, `Filter-Id`, `Session-Timeout`, `Idle-Timeout`, `Mikrotik-Recv-Limit` / `Xmit-Limit` / `Total-Limit` and gigawords variants. **Not** changeable: IP address, pool, routes.

Component structure



A RouterOS device configured as an IPoE BNG: core RouterOS on the left, the separately downloaded `user-manager` *package on the right, joined by the operator over RADIUS on the loopback.*

For IPoE, the authentication core is entirely bypassed. The DHCP service authenticates with a MAC in `User-Name` and an empty password; EAP, TLS, TOTP and method negotiation are inaccessible on that path.

Appendix B — Quantum Reference

The local AAA decision

```
resolve(locators, records, enforcement)
   → { subscriber_id, outcome, plan_name, framed_ip, timeouts, overlay }
```

Element	Values
Outcome	`accept` · `captive` · `reject`
Record status	`active` · `suspended` · `terminated`
Enforcement overlay	`throttle` · `captive` (and `clear` to remove)
Unknown-line default	`captive`
Default plan	`Bronze`
`aaa_source`	`open` · `local` (CE 1.4 default when never set: `open`; **1.5** makes it fail-closed `local`)

The REST surface

Route	Purpose
`GET` / `PUT /bng/aaa`	Source selection
`GET` / `PUT /bng/aaa/local`	`default_plan`, `unknown` policy
`GET /bng/aaa/local/subscribers`	List
`GET` / `PUT` / `DELETE …/subscribers/{id}`	Single record; PUT is a **replace**
`POST …/subscribers/bulk`	Atomic upsert to 10,000
`POST …/subscribers/bulk_delete`	Atomic delete
`POST …/subscribers/{id}/enforcement`	`throttle` | `captive` | `clear`, returns `applied_to_sessions`
`GET /bng/aaa/local/enforcement` · `POST …/enforcement/bulk`	Overlay read and mass apply
`GET /bng/subscribers/{id}/sessions` · `POST …/disconnect`	Subscriber-scoped view and concurrency trim
`GET /bng/accounting/records` · `POST …/records/ack`	CDR drain, default 500 / max 1,000
`GET /bng/counters` · `/bng/counters/{ip}`	Aggregate and per-session band counters
`GET /bng/sessions` · `/bng/sessions/{ip}` · `/bng/sessions/top` **[1.5]**	Session list, detail with `rate` and `history`, top talkers
`GET /bng/health` · `/quantum/license/status`	Counts, caps with provenance, verified grants

Defaults that matter operationally

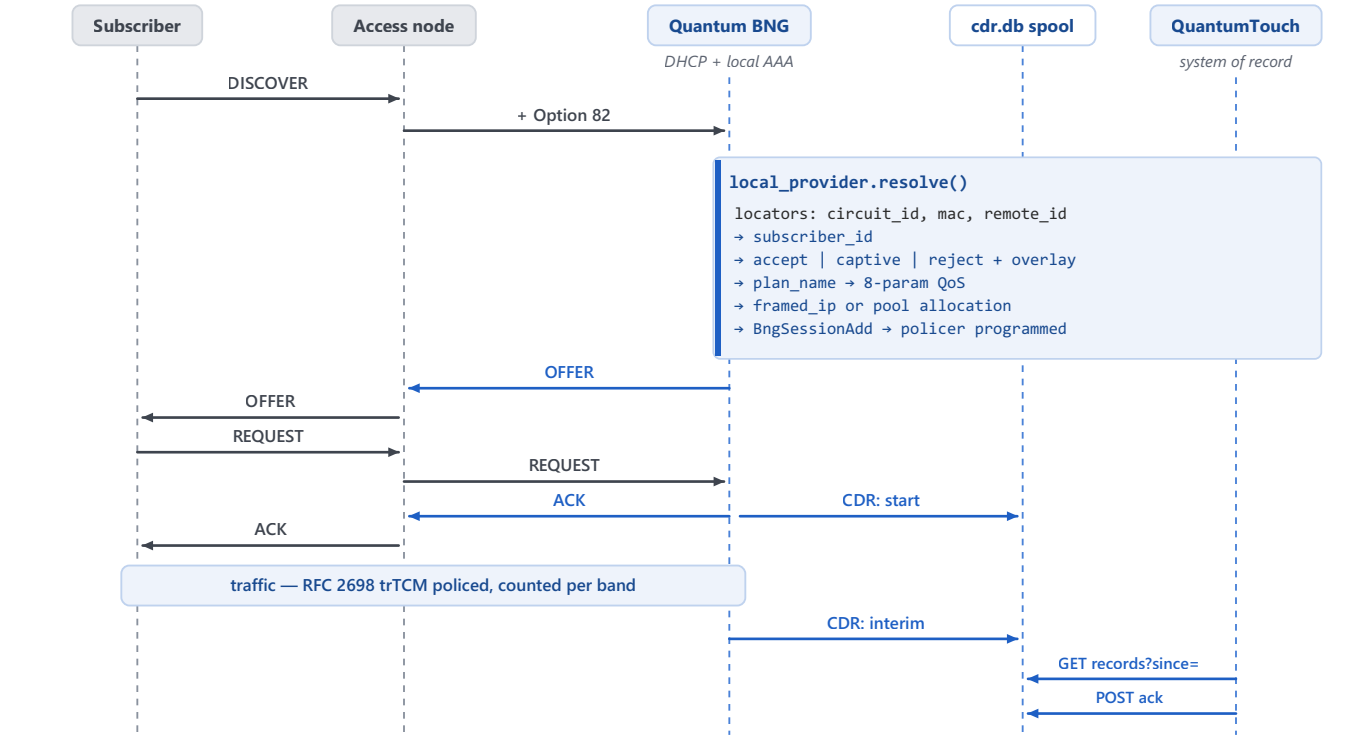
Default	Value	Rationale
`offer_timeout`	180 s	Spans common CPE retry gaps, so a still-retrying client is never reaped mid-onboard
`interim_default`	600 s	**RFC 2869 §5.16** ("SHOULD NOT be smaller than 600"); halves steady-state accounting load
Idle timeout	1 hour (raised from 5 min in 1.4)	Sized for always-on subscribers
Captive lease	120 s	Activation settles fast; the CPE renews and picks up real DNS
Telemetry cadence	5 s **[1.5]**	Independent of the accounting interim clock
History depth	120 samples ≈ 10 min **[1.5]**	Per warm session
Record timeout overrides	1–86,400 s	Per subscriber

A **plan** is eight integers: `dn_cir_kbps`, `dn_pir_kbps`, `up_cir_kbps`, `up_pir_kbps` (required) plus `dn_cbs_kb`, `dn_pbs_kb`, `up_cbs_kb`, `up_pbs_kb` — resolved to the data-plane policer at onboard.

Terminate causes — RFC 2866 numerics

`USER_REQUEST` 1 · `LOST_CARRIER` 2 · `IDLE_TIMEOUT` 4 · `SESSION_TIMEOUT` 5 · `ADMIN_RESET` 6 · `NAS_REBOOT` 11 · `SERVICE_UNAVAILABLE` 15

Onboarding call flow



A first-contact IPoE onboarding with `aaa_source = local`*. No RADIUS server appears anywhere in the exchange.*

Where Quantum goes next

Horizon	Item
Shipped, CE 1.4	Embedded local AAA, embedded DHCP, subscriber identity model, cascade, bulk to 10,000, static IP, reactive concurrency, durable CDR spool with per-band usage, reboot sealing, capacity tiers
In flight, 1.5	Fail-closed `aaa_source` default; node-prefixed fleet-unique `Acct-Session-Id`; CGNAT in the box; server-side session telemetry (top-N, per-band rate, history); 100,000-session tier
Optional connector	External RADIUS client, for operators with an existing RADIUS estate. The CDR spool is already the universal accounting substrate — the connector forwards from it rather than replacing it
Tracked gaps	Calendar-tiered prepaid; RadSec; arbitrary vendor attributes; external identity source; per-subscriber ACL policy

The connector point is the architectural one. The CDR client implements the same `accountingStart` / `accountingInterim` / `accountingStop` interface a RADIUS client would, but writes to a durable local spool instead of a wire protocol. The spool is then drained per edition — QuantumTouch tails it in Community Edition; a RADIUS client forwards from it commercially. Accounting durability is therefore a property of the platform, not of the operator's RADIUS server.

Sources

MikroTik primary documentation

- User Manager — RouterOS Documentation
- User Manager — RouterOS Manual
- RADIUS — RouterOS Documentation
- RADIUS — RouterOS Manual
- DHCP — RouterOS Documentation
- DHCP — RouterOS Manual
- RADIUS Client — RouterOS Docs mirror
- RouterOS license keys

Standards

- RFC 2698 — A Two Rate Three Color Marker
- RFC 2866 — RADIUS Accounting
- RFC 2869 — RADIUS Extensions
- RFC 5176 — Dynamic Authorization Extensions to RADIUS
- RFC 4679 — DSL Forum Vendor-Specific RADIUS Attributes

MikroTik field and community evidence

- Disconnect DHCP user from RADIUS?
- MikroTik DHCP RADIUS — Disconnect-NAK Unsupported-Extension
- 406 Error When Sending disconnect request to DHCP Radius
- New User Manager in RouterOS v7
- Session limit reached (current license allows only 200 session)
- TR101 support
- Edge Router & BNG Optimisation Guide for ISPs — Daryll Swer

---

Prepared by BlueDot Insight for the Quantum IPoE BNG launch. © 2026 BlueDot Insight LLC. BlueDot Proprietary.