MERCATURA / TECHNICAL FIELD GUIDE

Standard PQ vs. MercaWeave Grouped PQ

Witness V2 and Witness V3 - how they work, what changes, and when each is the right choice.



Designed for: exchanges, custody teams, mining pools, solo miners, wallet operators, integrators, and advanced users.

Core idea: V3 can dramatically reduce authorization overhead when many inputs are spent from the same group. It does not aggregate signatures from unrelated keys, and it is not universally better than V2.

VERSION 1.0 / 22 NOVEMBER 2026 / FOR TECHNICAL REVIEW BEFORE PUBLICATION

01 / FAST DECISION

Which one should you use?

Start with the spending pattern, not with the number of recipients. On Mercatura mainnet, both formats are supported from genesis, with distinct security and operational tradeoffs.

Default recommendation

Use **Standard PQ / Witness V2** unless you have a measurable, recurring need to spend many coins controlled by the **same MercaWeave group key**, plus the operational discipline to manage group-level risks.

Situation	Best starting choice	Why
Ordinary receive, save, occasional send	**V2**	Independent authorization and simplest default recovery/compatibility.
Exchange hot-wallet UTXO sweeping	**V3, selectively**	Large groups of same-key inputs can be consolidated with far less signature overhead.
Pool consolidating many reward UTXOs	**V3, selectively**	Repeated group-controlled rewards may share one signing authorization.
Pool paying many recipients from one input	**Often V2 is fine**	Output count does not itself create a V3 saving.
Solo miner taking occasional rewards	**V2**	V3 adds little if spends remain small or infrequent.
Cold storage / customer-segregated custody	**V2 first**	Avoid a larger shared-key compromise domain without a proven size benefit.
Cross-wallet or multi-party PSBT coordination	**Case by case**	V3 requires complete authenticated group/prevout coordination.

The key distinction

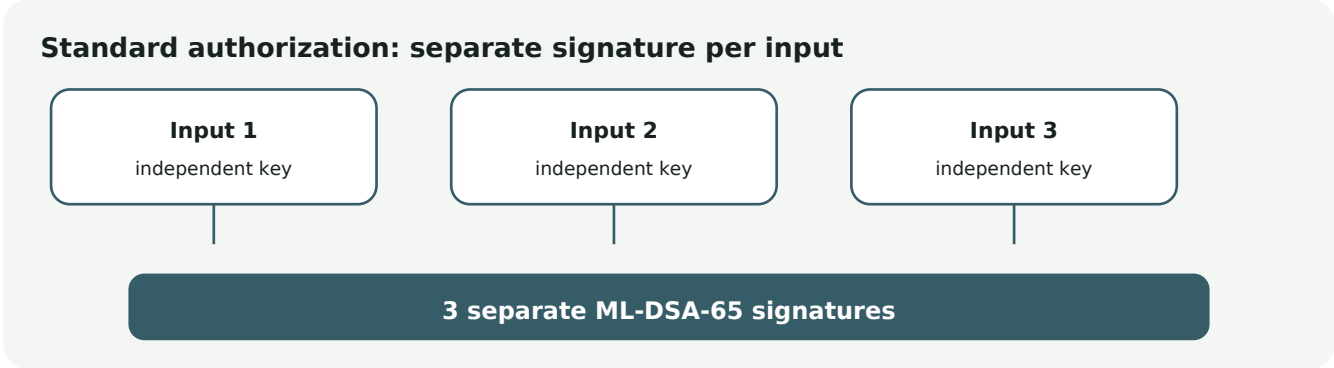
- **V2:** each spent input proves ownership with its own ML-DSA-65 signature and public key.
- **V3:** each group proves ownership with one ML-DSA-65 signature, while each member input still undergoes its own ownership/commitment checks.
- **Both:** use post-quantum ML-DSA-65; both remain subject to normal chain, fee, mempool, maturity, and wallet rules.

Security note: Mainnet enforces V3 from genesis. For custody, use compatible nodes and wallets; older clients that do not understand grouped PQ may misinterpret witness-v3 inputs. [S1, S2, S9]

02 / STANDARD PQ

Witness V2: independent authorization

The established native post-quantum spending path. V2 coins remain fully spendable alongside V3 coins; no conversion or migration is needed. [S2, S6]



How a V2 spend works

1. The PQ wallet derives a distinct child key under its existing private-only, network-bound derivation scheme.
2. A receiving output contains a native witness-v2 program committing to its ML-DSA-65 public key.
3. When spending, each V2 input carries its own public key and a transaction-bound signature; a verifier checks the commitment and authorization for that input.
4. Mixed inputs, multiple recipients, funded transactions, PSBT flows, and standard exchange withdrawal batching remain available. V2 does **not** mean one transaction per recipient.

Technical profile

Property	Witness V2
Cryptographic scheme	ML-DSA-65 (pure mode, Mercatura-specific authorization context).
Witness program	Version 2, exactly 32 bytes.
Per-input authorization	1,952-byte public key and 3,309-byte ML-DSA-65 signature (plus witness framing).
How signing scales	Typically one signature and one verification per V2 input.
Wallet default	Default native PQ receive, change and solo-mining destination.

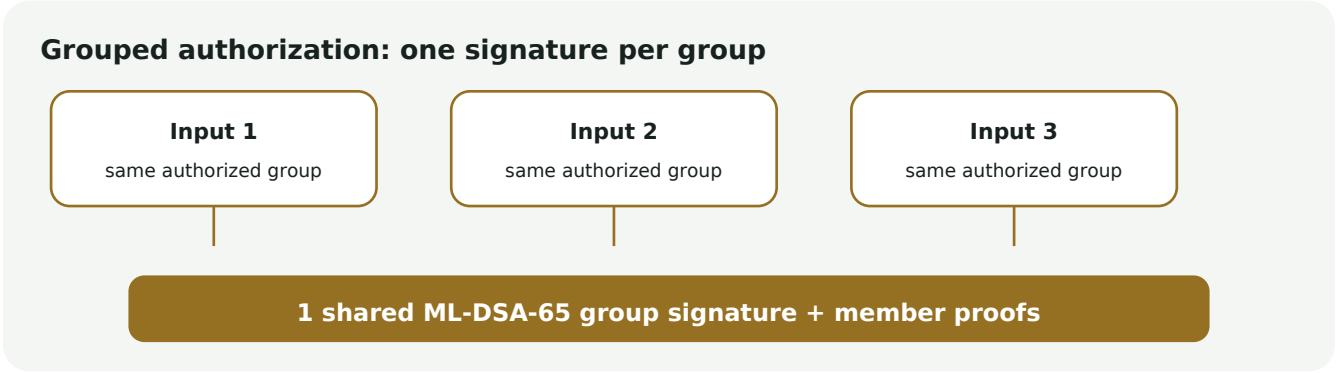
Benefits and costs

- **Benefits:** per-key compartmentalization, mature default workflows, no shared-group privacy coupling, straightforward mixed-input spending.
- **Costs:** large ML-DSA signatures repeat for each spent input, making high-input sweeps relatively expensive in transaction bytes and verification work.

03 / GROUPED PQ

Witness V3: MercaWeave authorization

MercaWeave is grouped authorization - not arbitrary cross-key signature aggregation. [S1, S2, S4, S6]



What changes inside a transaction

1. The wallet derives a **group key** (separated by network, account and purpose), then issues distinct 32-byte witness-v3 programs using authenticated per-address nonce/commitment data.
2. Inputs participating in a group carry canonical membership information. One designated **anchor** supplies the group public key and ML-DSA-65 signature; related **followers** provide compact authorization material.
3. One group signature authorizes the transaction-wide group manifest and the applicable spent-output data. Every member must still match its committed ownership and actual input context.
4. Different group keys require different signatures. A transaction with G groups and N grouped inputs does not collapse G separate keys into one signature.

Limits and work accounting

Rule	MercaWeave V3
Signature checks	One ML-DSA-65 verification per authenticated group, not per follower.
Group work formula	4G + N: G groups and N grouped inputs; independent native checks remain separate.
Grouped authorization bounds	At most 4,096 inputs, 256 groups and 1,024 members per group (plus ordinary block/policy limits).
Wallet issuance	The current candidate rotates to a new group after 64 issued addresses for each account/purpose.
Output destination	V3 uses a native 32-byte witness-version-3 program with compatible Mercatura wallet capability.

Best-case vs worst-case

Ten inputs in **one** group: one group signature. Ten inputs each from a **different** group: ten signatures and potentially a slight V3 overhead versus V2.

04 / COMPARISON

V2 vs. V3, side by side

Neither scheme is an unconditional replacement for the other.

Dimension	Standard PQ / Witness V2	MercaWeave / Witness V3
ML-DSA-65 authorization	One signature per V2 input	One signature per **group**; members separately checked
Small spend (1 input)	Simple, natural fit	Often similar or slightly larger
Large same-group spend	Size and signature cost grow with N	Signature cost grows mainly with G; compact followers
Many outputs, few inputs	Normal Bitcoin-family batching already supported	No automatic signature saving just from more outputs
Independent custody keys	Strong key-by-key compartmentalization	Multiple keys still mean multiple groups/signatures
Address privacy	No intrinsic shared-group identity	Related group inputs can be linked
Compromise blast radius	Limited to affected key and wallet secrets	Compromise of a group key threatens all unspent coins in that group
Wallet recovery	Established PQ wallet backup path	Requires authenticated issued-range/group metadata backups
Mixed spending	Can coexist with V3 in supported wallets	Compatible with V2 inputs in supported wallets
Coinbase rewards	Native default	Optional grouped reward destination
Mainnet consensus	Enforced from genesis	Enforced from genesis
Exchange/pool recommendation	Base default and fallback	Targeted optimization for same-group UTXO workflows

Three rules that prevent expensive mistakes

1. **Input count matters more than recipient count.** Payout batching with one funding input is already efficient in either format.
2. **Group cohesion matters.** A thousand UTXOs from many unrelated keys is not the same as a thousand UTXOs from a small number of securely managed groups.
3. **Custody boundaries matter.** Do not merge unrelated customer or reserve risk domains merely to save bytes.

Sources: [S1-S4]. This table covers authorization and operational properties, not a claim that either scheme is universally more secure.

05 / RECOMMENDED USE

How exchanges, pools and miners should deploy it

Operational best practice means using V3 where measured savings outweigh custody, metadata and compatibility costs.

Exchanges and custodians

- **Best fit:** high-input hot-wallet sweeps and treasury consolidations funded by coins issued into intentionally managed V3 groups.
- **Keep V2:** isolated custody, low-volume wallets, customer-specific segregation, and any reserve whose compromise must not propagate across grouped holdings.
- **Implementation:** maintain separate groups by security domain; create robust issued-range backups; benchmark actual funded transaction bytes; run PSBT coordinator with full prevout/role data; test replay, reorg and failed signing.

Mining pools

- **Best fit:** accumulating repeated coinbase or treasury UTXOs under appropriately rotated, authenticated V3 mining/receive groups, then sweeping many matured inputs at once.
- **Not a benefit by itself:** paying 500 miners in one transaction from a single pool wallet UTXO. This is output batching; V2 already supports it.
- **Important:** protect pool keys independently of exchange/operating reserves; respect coinbase maturity and fee-policy limits; use compatible node and wallet software for grouped PQ custody.

Solo miners and mining operations

- **V2 by default:** a single miner with infrequent rewards or small spends can retain the standard PQ wallet and payout path.
- **V3 selectively:** useful for high-volume reward accumulation and later same-group consolidation when strong backup and rotation discipline is in place.
- **No protocol change:** neither address type changes MercaHash mining, reward issuance, zero software-fee policy, or the ability to keep mining to V2.

Merchants, everyday wallets and long-term holders

- Prefer **V2** for simple receiving, payments, savings and cautious cold-storage separation. V3 is an opt-in optimization, not a mandatory migration.

Sources: [S2, S3, S7]. Recommendations above are operating guidance derived from documented properties, not formal exchange listing requirements.

06 / MEASURED RESEARCH

When does V3 actually save space?

The benchmark below reproduces official standalone research figures. They are illustrative signed-transaction byte comparisons, not live-chain fee quotes or production throughput claims.

Inputs	Groups	V3 serialized bytes	V2 adapter bytes	V3 byte reduction
1	1	5,397	5,364	**-0.62%**
10	1	6,117	53,145	**88.49%**
10	10	53,475	53,145	**-0.62%**
53	1	9,557	281,432	**96.60%**
53	10	56,915	281,432	**79.78%**
100	1	13,317	530,955	**97.49%**
100	10	60,675	530,955	**88.57%**
100	100	534,255	530,955	**-0.62%**

Benchmark provenance: WO01 standalone validation, one warm-up and five measured iterations per workload. V2 comparison bytes were produced through a research adapter using genuine ML-DSA signatures; the native production V2 script interpreter was not linked in these comparisons. No claim of production-standardness is implied. [S4]

Interpret the numbers correctly

- **53 inputs, one group:** roughly 96.6% lower transaction bytes in this research fixture, because the 1,952-byte public key and 3,309-byte signature are not repeated per input.
- **53 inputs, ten groups:** 79.78% lower bytes, because ten signatures are still substantially fewer than 53.
- **10 inputs, ten groups:** V3 is slightly larger; unrelated signing keys do not benefit from grouping.
- **Real fee impact:** depends on actual funded inputs, outputs, change, policy limits and the network fee rate. Mercatura fee sizing accounts for full serialized bytes rather than a Bitcoin-style witness discount.

Do not extrapolate from a headline percentage

A pool with 1 input and 500 payout outputs cannot assume a 96% fee saving. Grouped authorization primarily compresses repeated **input authorization**, not unrelated recipient outputs.

07 / CUSTODY & PRIVACY

Security, recovery and key rotation

V3 changes the operational fault domain even though it retains ML-DSA-65. Treat the group as a security boundary.

Security checklist

1. **Separate keys by risk domain.** Do not put unrelated customers, reserve balances and hot spending under the same group merely to share one signature.
2. **Rotate deliberately.** The native candidate rotates after 64 issued addresses in an account/purpose group, but UTXO count and repeated deposits can still grow independently.
3. **Plan group-key compromise.** One compromised group signing key endangers all remaining spendable coins controlled by that group. Use segregated hot/cold operations and migration procedures.
4. **Respect privacy leakage.** Grouped spends can correlate addresses/inputs sharing group authorization. Never promise the address independence of V2.
5. **Keep the required metadata.** Back up the encrypted PQ seed and durable authenticated V3 issued-range/group records and counters after new issuance.
6. **Test restores.** Seed-only recovery does not guarantee discovery of arbitrary unknown V3 issued ranges. After a cold encrypted restore, unlock/authenticate and rescan for missing inbound transactions.
7. **Prevent unsafe downgrade.** Mainnet enforces V3 from genesis. Old nodes treating v3 as unknown witness data may accept invalid spends. Wallets with required grouped records must not be opened by incompatible clients.

For PSBT and third-party coordination

- Require complete public input, spent-script/amount, group role, nonce and key metadata for every applicable participant; do not accept an unsupported proprietary field silently.
- Validate transaction-wide snapshot integrity after partial signatures and re-check UTXO membership, maturity and fees before broadcast. A valid group signature alone does not prove the coordinator described a live coin.
- Use authenticated ownership in the wallet signer; stage changes atomically; fail closed if entropy, signing, database or metadata validation fails.

Sources: [S1, S2, S5, S6]. The security summary reflects the project implementation and internal review; it is not an independent security certification.

08 / PRACTICAL WORKFLOWS

What users see in Mercatura Core Qt and RPC

Mercatura mainnet supports both PQ formats from genesis. Grouped features require compatible wallet software and authenticated group metadata. These are workflow examples.

Receive and choose a destination

Task	Expected supported behavior
Normal Receive / default	Standard PQ witness V2 remains selected by default.
Opt-in grouped Receive	Choose MercaWeave Grouped PQ (Witness V3) with a compatible private-key wallet and managed group metadata.
Simple RPC receive	`getnewaddress "label" "mercaweave" (wallet unlock as required).`
Advanced grouped RPC	`getnewmercaweaveaddress` supports label, account and receive/change/mining purpose.
Change policy	Ordinary/default change stays V2; grouped change is an explicit selection.

Send and coin selection

- **Automatic (recommended):** select eligible owned V2/V3 UTXOs and use normal coin selection, subject to ownership, maturity, fee and wallet-policy checks.
- **Standard PQ - V2 only:** fund from independent-key V2 coins; grouped coins are excluded.
- **MercaWeave - V3 only:** select authenticated, eligible grouped coins; V2 coins are excluded. Insufficient eligible funds can occur even when total wallet balance is higher.
- **Coin Control:** inspect input authorization type and group composition, compare a concrete funded byte/fee preview, and keep manual selections consistent with the spending mode.
- **RPC funding:** supported funding interfaces accept `spending_mode` = `"automatic"`, `"v2"` or `"v3"`. Funding-mode choice is independent of recipient output type.

A common mental model

The input pays; the output receives

A V2 input can pay a V3 recipient, and a V3 input can pay a V2 recipient. Spending mode governs the **UTXOs used for authorization**; it does not automatically convert the recipient or change address.

Sources: [S2, S3]. Method availability and exact RPC parameter schemas should be checked on the deployed version before integration.

09 / MAINNET OPERATIONS

Mainnet operations and integration checklist

Recommended minimum checks for an exchange, pool, miner, or public Explorer integration.

1. **Confirm mainnet software.** Standard PQ and MercaWeave Grouped PQ are enforced from genesis. Verify that validators, mining nodes and wallets recognize and validate witness-v2 and witness-v3 spending with compatible Mercatura Core software. [S1, S9]
2. **Keep validators and miners current.** Older software may treat witness-v3 inputs as unknown witness programs; avoid incompatible nodes for mainnet custody or mining.
3. **Rehearse critical failure scenarios.** Test invalid witness roles, cross-group signers, reorgs, restarts, mempool persistence, interrupted signing and wallet recovery.
4. **Do a real custody restore.** Exercise encrypted wallet backups, post-issuance range records, required wallet-feature handling, unlock, rescan and signing after restart.
5. **Measure real fee savings.** Compare actual signed mixed-input and V3-only transactions at equal fee rates and equivalent outputs; enforce standard transaction weight and other limits.
6. **Coordinate pools and third parties.** Confirm address issuance, PSBT support, signing policy, grouped role metadata, payout maturity, reorg behavior and conservative fallback to V2.
7. **Document failure paths.** Handle locked wallets, unsupported software, depleted eligible V3 inputs, lost issued-range metadata, key compromise and unexpected service pauses.
8. **Maintain production readiness.** Confirm compatible nodes, wallet versions, monitoring, recovery procedures, operator training and incident-response processes.

FAQ

Does V3 replace V2? No. Existing V2 coins remain valid; V3 is optional and no automatic migration is required.

Does V3 make 1,000 payouts cheap? Not merely because there are 1,000 recipients. The major optimization is when many funding UTXOs are members of a small number of groups.

Does V3 require one seed per group? The candidate derives groups from the wallet's protected PQ master seed using a separate versioned group domain and authenticated issuance metadata.

Are V3 transfers intrinsically more quantum-resistant? Both schemes use ML-DSA-65. V3 improves grouped authorization efficiency, not the ML-DSA parameter set.

Should a pool immediately switch its coinbase rewards to V3? Only after evaluating custody boundaries, grouped-key backups, rotation and repeated-UTXO spending needs. V2 remains the safe default.

10 / SOURCE NOTES

Evidence and publication notes

This guide is a practical interpretation of Mercatura Core project sources, not a substitute for consensus specification, reviewed releases, or an external security assessment.

Source basis

Ref.	Mercatura source (project repository)
[S1]	**MercaWeave consensus integration and deployment** (open source)
[S2]	**MercaWeave wallet and Qt compatibility** (open source)
[S3]	**Native Send spending modes** (open source)
[S4]	**MercaWeave standalone validation and benchmark table** (open source)
[S5]	**MercaWeave final internal security qualification** (open source)
[S6]	**MercaWeave source audit - authorization and cryptography** (open source)
[S7]	**Candidate build, review and release preparation** (open source)
[S8]	**MercaWeave engineering completion and owner handoff** (open source)
[S9]	**Mainnet genesis consensus configuration** (open source)

Evidence labels

Source-grounded: protocol sizes, wallet interfaces and research fixture results. Inferred operational **guidance:** who should prefer V2 or V3. Not independently verified here: deployed binary integrity, external security assessment or live-chain fee savings.

Short glossary

Term	Meaning
UTXO	Unspent transaction output; a spendable coin selected as a transaction input.
PQ	Post-quantum cryptography; here ML-DSA-65 digital signatures.
Witness V2	Mercatura native PQ independent-key authorization format.
Witness V3	MercaWeave grouped PQ authorization format.
Anchor	Member that supplies the one signature/public-key authorization for its group.
Follower	Additional group-owned input with compact per-member authorization data.
PSBT	Partially Signed Bitcoin Transaction container, adapted by Mercatura for supported PQ workflows.

Prepared: 22 November 2026. Source reviewed: Mercatura-Core/Mercatura-Core (project sources published through November 2026). Recheck all exact network parameters, RPC APIs, and release identities before public distribution.