NetWrap: A Deficiency-Resilient Wrapper for Multilateral Netting

Daniel Aronoff
Massachusetts Institute of Technology

Draft date: October 14, 2026

Abstract

NetWrap adds verifiable prefunding and deterministic recovery to any multilateral netting core. If a participant cannot fund, it removes only deficiency-linked portions of obligations, restores them bilaterally, re-nets the remainder, and settles from locked objects. Across epochs, locks persist, adjust, and support terminal settlement without changing original counterparty relationships.

1. The design problem

Multilateral netting reduces the objects that must move at settlement. The arithmetic is simple: offset each participant's inflows and outflows for each object type, then transfer only the net amounts. The institutional problem is harder. A netting method also determines who bears the loss if a required transfer is not funded. That second effect can be more important than the reduction in settlement volume.

Current methods make different choices. Trade compression cancels offsetting contracts and replaces them with fewer contracts, which can create a direct exposure between parties that did not originally trade. Central clearing preserves multilateral netting by interposing a central clearing counterparty, which replaces the original exposure map and generally supports the new map with margin, default funds, and other loss-allocation rules. Atomic cycle methods preserve the original contracts, but a single failure can prevent an otherwise feasible cycle from settling. Liquidity-saving mechanisms select a feasible subset and defer the rest, but ordinarily do not convert the deferred portion into a bounded, endpoint-preserving recovery process.

Method	What is netted	Response to a shortfall	Counterparty consequence
Trade compression	Gross contracts	Replaces the contract set	May create new bilateral exposures
Central clearing	Obligations novated to a CCP	CCP risk-management waterfall	Replaces bilateral exposure with CCP exposure
Atomic cycle or feasible subset	Selected instructions	Aborts or defers the infeasible set	Preserves endpoints, but may sacrifice execution
NetWrap	Any chosen netting core	Cure, selective removal, bilateral reinstatement, and re-netting	Preserves original endpoints for the unfunded portion

The governing distinction is therefore between a **netting core** and an **execution-and-deficiency-handling layer**. The netting core computes the offsets. The second layer decides when the result may execute and what happens if a participant cannot establish the required funding state. NetWrap operates at this second layer. It does not require a new netting algorithm.

2. The architecture

A coordinator node, called the mechanism operator (MO) here, executes NetWrap as a computing process. Participants are endpoint nodes; their obligation records represent initial scheduled transfers. The MO is not a trading counterparty and does not guarantee the obligations. It receives authenticated scheduled transfer records, assigns transfers or portions to coordination groups, invokes the selected netting core (the group construction method), and creates a versioned multilateral coordination record (MCR) for each group. An MCR states each participant's required net object outflow and net object inflow per object type, called net outflow and net inflow below. A coordination group contains transfers or portions. The MCR governs execution; underlying bilateral agreements may remain in force.

Before settlement, every participant with a required net outflow must place that quantity into a verifiable locked state. A bank hold, custodian escrow, smart-contract escrow, or another authoritative reservation system can supply the lock. The receipt identifies the protocol run, MCR version, object type, quantity, and commitment window. Settlement instructions can use only quantities attributed to the current terminal MCR.

The protocol then follows five steps:

1. Compute per-participant net positions and issue the current MCR.
2. Lock and verify every required net outflow.
3. If a shortfall exists, apply any verified cure supplied by an authorized external party having cure authority.
4. If a deficiency remains, remove only deficiency-linked portions of the deficient participant's sending assignments, restore those portions as bilateral obligations between the original counterparties, and re-net the residual group.

5. Repeat recovery until the current MCR is fully committed—all required net object outflows are verifiably locked. If the settlement-control indicator permits immediate or final execution, settle only from those locked objects; otherwise retain the locks and continue to the next epoch without distribution.

Figure 1 shows the distinction. The selected netting core remains responsible for the arithmetic. NetWrap controls commitment, deficiency recovery, and settlement.

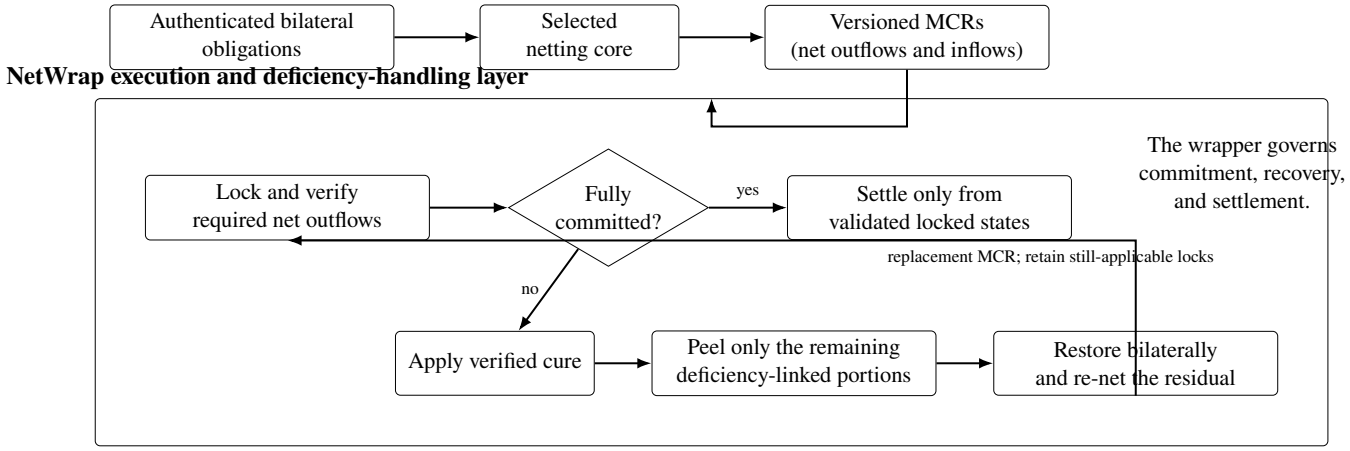


Figure 1: NetWrap leaves the netting arithmetic modular and changes the response to a funding shortfall. Adapted from the protocol architecture in Figures 2 and 13 of the technical paper.

The persistent state is equally important. The MO maintains a scheduled transfers table, a versioned MCR store, a commitment ledger, a deficiency log of removed portions and reinstated bilateral transfers, and an append-only transcript ledger. Every transition is bound to the run and version. A late receipt cannot fund a superseded MCR, and replay after interruption cannot apply the same settlement instruction twice.

3. A single-epoch run

Let q_{ij}^k denote the quantity of object k that participant i must send to participant j within a coordination group. Participant i's signed net outflow is

$$n_i^k = \sum_j q_{ij}^k - \sum_j q_{ji}^k.$$

The MCR therefore assigns

$$R_i^k = \max(0, n_i^k), \qquad P_i^k = \max(0, -n_i^k),$$

where R is the required net outflow and P is the required net inflow. The total required outflow equals the total required inflow for each object type.

Suppose the eligible verified locked quantity attributed to this participant, object type, and current MCR, after any allocation among groups and before the cure below, is L_i^k. At the commitment-window deadline, the detected deficiency is

$$\delta_i^{k,0} = \max(0, R_i^k - L_i^k).$$

If an authorized external party supplies a cure whose resulting lock is verified B_i^k, only the remaining deficiency enters the recovery rule:

$$\delta_i^k = \max(0, \delta_i^{k,0} - B_i^k).$$

A full cure produces no removal. If δ_i^k is positive, the MO selects one or more sending assignments of i. For a selected assignment e having quantity q_e, the MO may retain a fraction α_e in the coordination group and remove the balance. The selected portions satisfy

$$\sum_e (1 - \alpha_e) q_e = \delta_i^k,$$

where exact equality is feasible. If exact removal is infeasible under the object's minimum transferable unit (MTU), the smallest quantity permitted by its custody system, the MO removes the smallest permissible quantity not less than the deficiency and records the overshoot and deterministic tie-break.

The removed portion is not canceled economically. It is reinstated as an equal-quantity bilateral scheduled transfer of the same object type between the original sender and receiver, and is excluded from settlement under the residual MCR. The residual group is then re-netted. This point separates NetWrap from an ordinary deletion rule. Deletion can make the unpaid claim disappear from the settlement process. NetWrap changes its settlement path while preserving its endpoints.

A compact example

Consider one object and four participants with the following obligations:

$$A \to B : 10; \quad B \to C : 10; \quad C \to A : 6; \quad D \to B : 4.$$

Before any deficiency, the net positions are:

Participant	Required net position
A	Out 4
B	In 4
C	In 4
D	Out 4

The multilateral settlement therefore moves eight units instead of thirty. Suppose D locks only one unit. D has a deficiency of three. NetWrap removes three units from D's four-unit obligation to B and restores those three units as a bilateral D-to-B obligation. The residual group contains D-to-B of one unit. Its net positions are A out 4, D out 1, B in 1, and C in 4. Five units settle multilaterally from verified locks; three units remain a bilateral obligation between D and B.

Only D and B change when the residual group is re-netted. A and C were not endpoints of the removed portion, so their net positions remain unchanged. More generally, removing an assignment can alter only the positions of its two endpoints. This locality permits the MO to carry forward unaffected positions rather than recompute every node in a large group.

Each unsuccessful recovery iteration removes a positive quantity from the active coordination group. For indivisible units, the number of active minimum transferable units is a non-negative integer that strictly falls. For whole-transfer recovery, the number of active transfers falls, provided each iteration removes at least one and none is readmitted within the run. These integer potentials give a finite bound for a fixed run under deterministic tie-breaking. Unrestricted continuous fractional peeling needs a sufficient single pass or another validated finite-progress procedure.

4. Multiple epochs

A repo, derivative, securities loan, or other continuing agreement can generate obligations at several dates. The central distinction is between **obligation state** and **funding state**. The agreement and the committed market observation determine the current obligation. A lock established at an earlier epoch is existing funding applied only after that obligation has been calculated. Treating the lock as an input to the obligation formula would count the same object twice.

At a non-final epoch, a fully committed MCR can serve as a lock-requirement record rather than a settlement instruction. The required quantities remain unavailable for reuse. The next epoch's MCR supersedes the earlier record, but it need not release and re-establish the quantities that remain required.

Let $L_{i,k,m-1}$ be the valid continuing lock entering epoch m and $R_{i,k,m}$ the updated required outflow. The MO computes

$$\begin{aligned} C_{i,k,m} &= \min(L_{i,k,m-1}, R_{i,k,m}), \\ E_{i,k,m} &= \max(0, L_{i,k,m-1} - R_{i,k,m}), \\ I_{i,k,m} &= \max(0, R_{i,k,m} - L_{i,k,m-1}). \end{aligned}$$

It carries C forward without release, releases the excess E, and requests and verifies only the increment I. In this release-excess embodiment, after I is verified, the non-final conservation relation is

$$L_{i,k,m-1} + I_{i,k,m} = E_{i,k,m} + L_{i,k,m}.$$

The provisional net inflows do not settle. At the final epoch, the continuing and incremental locks fund the terminal MCR.

Figure 2 gives the intuition. A requirement of 10 rises to 13, falls to 8, and ends at 9. The MO locks 10, adds 3, releases 5, and adds 1. It never asks the participant to release and re-lock the unchanged portion.

A recurrence round still needs controls. Obligations must share an epoch and observation cutoff; both counterparties must affirm amount, object type, direction, and epoch; object elections must be determined before netting; and the transcript must bind each

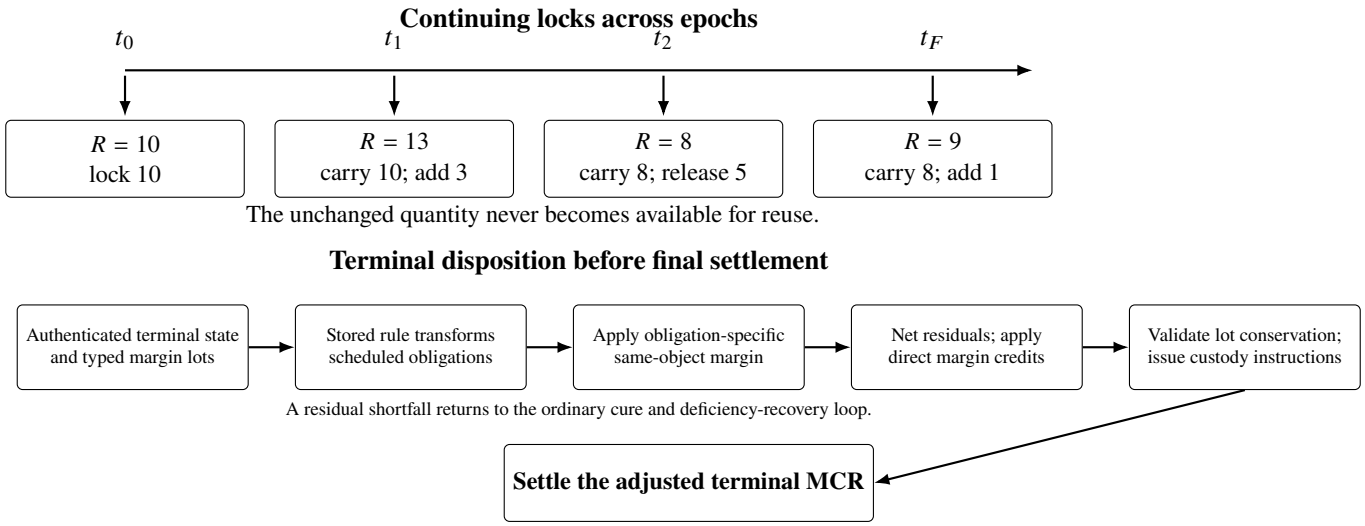


Figure 2: NetWrap carries only the required lock across epochs, then resolves terminal margin before final settlement. Adapted from Figure 14 and the terminal-margin extension in the technical paper.

new epoch to the prior transcript root and lock-state digest. A continuing quantity cannot support another current MCR or another deferred-settlement sequence at the same time.

Where every admitted obligation satisfies a homogeneous common recurrence rule on a verified matching source-aggregation basis and object-type mapping, the stored net-position vector can be multiplied by the common factor; a negative factor reverses direction. Reuse requires both the common-factor test and the complete matching-basis test. If either test fails or cannot be verified, the MO recomputes from the affirmed admitted obligations or uses a separately verified matching aggregate.

5. Terminal margin and final settlement

Continuing margin creates a separate problem at the terminal epoch. A locked quantity may fund the posting party's scheduled payment, return to the posting party after performance, transfer to a protected counterparty after nonperformance, remain held during a dispute, or be converted under an express rule. The mere existence of margin does not determine which result applies. The governing agreement and an authenticated terminal performance-state record do.

NetWrap represents each terminal-support quantity as a margin lot: a machine-readable record of a specified locked object quantity available for terminal settlement. The record identifies the object type, quantity, custody lock, posting and return nodes, eligible obligation or set-off scope, protected party where applicable, allocation priority, and terminal-disposition rule. The MO partitions a lot among obligation-specific applications, margin credits to endpoint nodes, continued holds, and any expressly authorized conversion input. For a lot ℓ of quantity m_ℓ,

$$\sum_e A_{\ell e} + \sum_i C_{\ell i} + H_\ell + X_\ell = m_\ell.$$

The validator rejects a proposed transition if the active and pending dispositions would exceed the source lot or if the same lot portion would be active in more than one disposition. Only after successful validation and atomic commitment of the dispositions, source-lot state, and any pending instruction record may the corresponding custody instruction be issued.

The order of operations matters:

1. Authenticate final tenders, locks, deliveries, defaults, and dispute states.
2. Apply the stored agreement rule to determine which scheduled legs survive, which are withheld, and which replacement or close-out obligations arise.
3. Apply obligation-specific margin to eligible obligations of the same object type.
4. Net the residual obligations by object type.
5. Apply scoped same-object-type margin credits against the credited participant's residual net outflow, retaining their lock without double counting.
6. Run the ordinary cure and deficiency-recovery loop for any remaining shortfall, then settle.

Cash margin cannot itself deliver a security. If failure to deliver a security creates a cash close-out claim under the agreement, cash margin can satisfy that cash claim. Cross-object use otherwise requires an explicit conversion record stating the valuation

source, observation time, haircut, rounding rule, and custody operation.

Terminal repo example

Let x be cash, y one Treasury security, and z another security. At the terminal date:

- e_1: A pays B $100x$, and B returns $10y$ to A.
- e_2: B pays C $80x$, and C returns $8z$ to B.
- e_3: C pays A $50x$, and A delivers $5y$ to C.

A has a $20x$ settlement-margin lot that may fund A's cash payment. B has a $12x$ performance-margin lot that returns to B if B delivers $10y$ and may support a cash close-out claim to A if B fails.

If all parties perform, the cash legs net to A out $50x$, B in $20x$, and C in $30x$. A uses its $20x$ lot and must add only $30x$. B's $12x$ lot returns to B.

Suppose B does not produce the authoritative tender receipt for $10y$. The stored atomicity rule withholds both legs of e_1, and the stored close-out rule creates B-to-A of $15x$. The MO applies B's $12x$ lot, leaving a residual B-to-A obligation of $3x$. A's own $20x$ lot returns because A's $100x$ leg did not settle. The remaining cash obligations are B-to-C $80x$, C-to-A $50x$, and B-to-A $3x$. After netting, B owes $83x$, C receives $30x$, and A receives $53x$. A also receives $12x$ from B's margin and the return of its own $20x$.

The example shows three distinctions. First, the agreement determines the close-out obligation; the margin lot only funds it. Second, obligation-specific margin is applied before residual netting, while returned or general credits are applied after netting. Third, an excess lot is returned or held unless an additional penalty or forfeiture obligation expressly authorizes its transfer.

6. What the protocol establishes

Fully committed settlement. A current immediate- or final-epoch terminal MCR executes only when eligible verified locked quantities, including valid margin credits and verified cures counted once, cover every required net object outflow and its settlement-control indicator permits execution. Settlement does not rely on an inventory assertion made before the commitment deadline.

Endpoint preservation. A deficiency-linked portion that leaves a coordination group is restored between the original sender and receiver. NetWrap does not create a new unsecured counterparty for that portion. A close-out, novation, or new trade can change the exposure graph, but those are contractual events rather than effects of the recovery algorithm.

Net-quantity and economic-entitlement invariance. Per participant and object type, signed residual-MCR settlement plus signed reinstated bilateral quantities equals the initial signed net quantity. Under full performance, this preserves the corresponding net economic entitlement; it does not preserve every gross delivery or receipt. Unperformed bilateral obligations may still affect realized profit.

Minimum recipient delivery for fixed net positions. For each object type, recipient deliveries under the MCR equal aggregate net object inflow for the surviving group. This minimum concerns net settlement deliveries, excludes custody-funding and intermediary movements, and does not assert globally optimal transfer or removal. Unfunded portions remain bilateral and do not block settlement of the funded remainder.

Finite recovery for a fixed run. Every non-terminal peel removes a positive amount from the active group. Under the stated integer-potential conditions, a sufficient single pass, or a validated procedure that eliminates deficiencies without creating another, recovery has a finite bound verifiable from the transcript. A positive real-valued decrease alone is insufficient.

Cross-epoch conservation and no double use. A continuing lock is either carried, released, incremented, or finally distributed. A terminal margin lot is partitioned among mutually exclusive uses. The commitment ledger and invariant validator prevent concurrent attribution of the same quantity.

Deterministic replay. Run identifiers, MCR versions, message identifiers, signed receipts, and a hash-linked transcript permit a restarted MO or an authorized reviewer to reconstruct the same state and suppress stale or duplicate instructions.

7. Implementation and limits

NetWrap can operate over institutional accounts or smart contracts. In an institutional implementation, the MO sends lock, release, and transfer instructions to banks, custodians, central securities depositories, or collateral managers and receives authoritative receipts. In a blockchain implementation, smart-contract escrow records the locks and settlement transfers. Heterogeneous rails can be joined through adapters, although cross-rail finality may require time-bounded locks, staged release, or another conditional-commit mechanism.

Privacy and governance are separate implementation choices. The MO may execute inside a trusted execution environment, through secure multiparty computation, or with zero-knowledge proofs of selected invariants. A single operator or a distributed set of operator nodes can maintain the state. An authorized external party may review a scoped transcript and, if separately granted cure authority, supply a cure. None of these choices changes the basic sequence of lock, verify, cure, peel, re-net, and settle.

The same architecture applies to non-financial entitlements that an authoritative controller can reserve and transfer. Network bandwidth slices, quality-of-service priority tokens, cloud-compute capacity, and logistics slots can serve as object types. A software-defined-networking controller or quota manager can supply the lock and transfer receipts. The example matters because NetWrap is a distributed commit and recovery protocol. Finance is its principal application, not a limitation of its control logic.

NetWrap does not price a derivative, calculate a repo haircut, determine eligible collateral, decide whether a close-out right is enforceable, or eliminate the original bilateral counterparty exposure. Those rules come from the governing agreements, market infrastructure, and applicable law. The protocol assumes authoritative object records, authenticated observations, enforceable settlement instructions, and a defined response to disputed performance. It makes the resulting state transitions explicit, bounded, and auditable.

8. The result

Existing netting methods tend to choose between execution and exposure preservation. Compression and central clearing obtain netting by changing the exposure map. Atomic methods preserve the map by refusing to execute an infeasible set. NetWrap changes the deficiency-handling rule. It prefunds the multilateral result, removes only what cannot be funded, restores that portion to its original bilateral path, and continues with the remainder.

The same logic extends across time. Continuing locks fund changing requirements without repeated release and re-establishment. At the terminal epoch, authenticated terminal performance-state records and terminal-disposition rules determine how locked quantities are applied, returned, held, or converted. The wrapper therefore separates three questions that incumbent systems often combine: what the parties owe, what the netting core offsets, and what the computer system may safely execute.

Source note. This overview summarizes the NetWrap technical paper, including its multi-epoch and terminal-margin extensions. The full paper contains the normative protocol, proofs, persistent-state model, and implementation maps.