

Available-to-Sell Reservations Under Network Partitions: Contract Limits, Escrow Safety, and Risk-Aware Rights Allocation

Pedro Alberto Laris Uribe
Independent Researcher
pedro.laris.uribe@gmail.com
github.com/peyidev/inventory-cap-simulation

Abstract

The phrase “real-time inventory” commonly conflates physical-stock accuracy, event-capture latency, and replicated-state semantics. This paper isolates the operational primitive of available-to-sell (ATS) reservation. We specify an idempotent sequential ATS object and separate three operational properties: a global commitment budget, termination at every non-failing replica, and no false rejection in quiescent histories. A local-history indistinguishability proof shows that no implementation can preserve all three when a partition separates non-failing replicas; strict linearizable ATS is therefore impossible under replica availability. We then instantiate the established bounded-counter escrow construction as an ATS object. The specification includes an atomic durable request-outcome map, an inductive conservation proof, quiescent convergence, recovery and fencing conditions, and a sufficient barrier protocol for exact global queries. Protocol safety is separated from physical-stock uncertainty. We give a risk-composition bound and a finite-sample, distribution-free buffer from exchangeable audit errors. For rights placement, classical discrete marginal allocation specializes to demand-tail probabilities; mean-proportional allocation has no constant approximation guarantee as replica count grows. A reproducible model experiment contains 60,000 run-level records across twelve scenarios. Common-random-number pairings quantify partition exposure, rerouting, forecast error, and leader placement; the mixed-quantity policy uses exact compound-Poisson marginals. Results are restricted to the declared models: stale AP accepts excess commitments, fixed-leader CP depends on reachability and leader location, and escrow preserves the budget but can reject because rights are fragmented or misallocated. Source, reference outputs, and validation scripts are maintained at <https://github.com/peyidev/inventory-cap-simulation>.

Keywords: available-to-sell, inventory reservation, CAP theorem, linearizability, escrow, bounded counter, CRDT, distribution-free calibration, resource allocation

1 Introduction

Omnichannel systems make stock commitments at customer-interaction time: display availability, accept a basket, reserve units, and route fulfillment. A common requirement states that every channel must immediately see the same stock and that

every reachable channel must continue selling during failures without overselling. That sentence combines three different concerns:

1. **Physical truth:** how many units can actually be picked.
2. **Capture latency:** how quickly receipts, sales, shrinkage, and audits enter the digital record.
3. **Replicated-state semantics:** which operations complete and which invariants hold when replicas cannot communicate.

A protocol can solve the third problem without making the recorded count physically correct. Conversely, better sensing cannot make contradictory distributed semantics implementable.

This paper isolates ATS reservation. It does not claim a complete order-management or inventory-lifecycle protocol. Receipts, expiration, cancellation, returns, multi-SKU atomicity, and negative corrections remain explicit boundaries.

The complete source, run-level data, paired comparisons, tests, metadata, checksum manifest, and manuscript are maintained in the canonical repository <https://github.com/peyidev/inventory-cap-simulation> [1]. Artifact version 3.1.0, `ARTIFACT_IDENTITY.json`, and `SHA256SUMS` identify the exact archived file contents used for every reported value; the hosting Git commit identifies the uploaded repository copy.

1.1 Contributions and claim discipline

The technical contributions are:

- independent safety, liveness, and opportunity properties for ATS, plus an inventory-local indistinguishability proof more general than the strict-linearizability corollary;
- an escrow ATS specification with atomic durable idempotency, a local inductive invariant, a global no-oversell theorem, quiescent convergence, and explicit recovery, fencing, garbage-collection, and query boundaries;
- a two-layer physical-risk composition and a distribution-free audit-calibration theorem requiring no Gaussian error model;
- an ATS specialization of discrete marginal allocation, a fixed-component extension, and a counterexample showing that mean-proportional placement can be arbitrarily poor as replica count grows;
- a paired reproducible model evaluation covering partition exposure, replica count, rerouting, asymmetric demand, forecast

reversal, leader placement, and non-unit quantities.

CAP impossibility, escrow transactions, bounded-counter CRDTs, reservation-based disconnected operation, invariant-preserving weak consistency, marginal resource allocation, and conformal/order-statistic calibration are established foundations. Table 1 distinguishes inherited results from this paper’s synthesis and extensions. The paper does not claim a new consensus protocol, CRDT family, universally optimal dynamic rebalancer, production-calibrated stock-error model, or database-product benchmark.

2 System Model and ATS Contracts

2.1 Sequential object

Definition 2.1 (Idempotent sequential ATS). Let $ats \in \mathbb{Z}_{\geq 0}$ be initialized to an allocation budget $Q \in \mathbb{Z}_{\geq 0}$. A request identifier x is globally unique, and $q \in \mathbb{Z}_{>0}$.

- $\text{Query}() \rightarrow v$ returns $v = ats$.
- For unseen x , $\text{Reserve}(x, q)$ returns OK and applies $ats := ats - q$ if $ats \geq q$; otherwise it returns INSUFFICIENT without changing ats . The pair $(q, \text{outcome})$ is recorded for x .
- A duplicate $\text{Reserve}(x, q)$ returns the recorded outcome and performs no second transition. Reuse of x with a different quantity returns INVALID.

Q is an allocable commitment budget, not necessarily recorded on-hand or physical stock. Section 5 relates those quantities.

2.2 Network, failures, and fairness

There are $k \geq 2$ replicas in an asynchronous message-passing system. Messages can be delayed arbitrarily. A partition is an interval in which messages between some non-failing replicas are not delivered. The impossibility proof uses crash-stop failures. The escrow object uses crash-recovery replicas with durable single-writer state.

A replica’s local history contains invocations, responses, local steps, completed durable writes, received messages, local clock readings, and realized random choices. Clock values have no assumed relation to a global clock or bounded message delay; the history contains no unreceived remote event.

Assumption 2.1 (Fair admissible execution). A non-failing replica is scheduled weakly fairly: a continuously enabled local step eventually executes, and a local durable write issued by that replica eventually completes. During a partition, no delivery fairness is assumed across disconnected components. After communication is restored, retransmitted messages between non-failing replicas are eventually delivered.

2.3 Independent operational properties

Definition 2.2 (Global budget safety). At every finite execution prefix, the sum of quantities associated with distinct fresh request identifiers that returned OK is at most Q .

Definition 2.3 (Replica availability). For every fair admissible execution, each operation invoked at a non-failing replica eventually returns without requiring delivery from a currently unreachable replica.

Definition 2.4 (Quiescent opportunity completeness). If a completed fresh $\text{Reserve}(x, q)$ overlaps no other fresh reservation and the successful reservations completed before its invocation total at most $Q - q$, it returns OK rather than a false refusal.

These properties are independent: safety limits accepted quantity, availability is liveness, and opportunity completeness forbids needless rejection in a nonconcurrent history. They expose the operational trade-off without treating a consequence of linearizability as a separate axiom.

Definition 2.5 (Strict ATS contract R4). R4 requires Definition 2.1 to be linearizable and replica-available in every fair admissible execution, including partitioned executions. Linearizability places each completed operation between invocation and response, respects real-time precedence, and produces a legal sequential history [5].

R4 implies Definitions 2.2 and 2.4. It also forbids UNAVAILABLE, timeout, or false INSUFFICIENT as substitutes for the sequential result.

3 Partition Impossibility

Theorem 3.1 (Safety–availability–opportunity impossibility). *No implementation satisfies global budget safety, replica availability, and quiescent opportunity completeness in every fair admissible execution of an asynchronous system that admits a partition between two non-failing replicas.*

Proof. Let $Q = 1$ and let a partition isolate replicas A and B . In execution E_0 , no operation occurs at A , and a client invokes a fresh $\text{Reserve}(x_B, 1)$ at B . Replica availability forces completion after a finite local history h_B . Because no successful reservation precedes the invocation and no reservation overlaps it, opportunity completeness requires OK.

Construct E_1 so that a client first invokes fresh $\text{Reserve}(x_A, 1)$ at A . Availability forces completion, and opportunity completeness requires A to return OK. After that response and before the invocation at B , the partition hides A ’s operation from B . The environment gives B the same invocation and schedules the same local steps, completed durable writes, received messages, clock readings, and realized random choices as in h_B .

Replica B therefore returns OK in both executions. Execution E_1 has two distinct successful unit reservations and violates global budget safety. For a randomized implementation claiming correctness for every execution, fix the realized choices in the identical local histories. $\square$

Corollary 3.2 (R4 impossibility). *No implementation of Definition 2.1 satisfies R4 when a partition can separate two non-failing replicas.*

Table 1: Claim map. Established results are used as foundations rather than relabeled as new algorithms.

Ingredient	Established basis	Contribution in this paper
Partition impossibility	Linearizability and CAP impossibility [5, 2]	Independent ATS safety, replica-termination, and quiescent-opportunity properties; an inventory-local-history proof with explicit fairness assumptions.
Escrow and reservations	Escrow, leased reservations, and bounded counters [6, 7, 8, 9]	Reservation API with atomic home-idempotency state, inductive quantity conservation, non-regressing recovery, fencing and request-retention boundaries, and exact-query conditions.
Invariant-aware execution	Coordination avoidance, predictive treaties, and invariant-preserving weak consistency [11, 13, 14, 15, 16, 17]	ATS-specific contract decomposition and explicit separation of system-budget safety from physical-stock risk.
Physical uncertainty	Inventory-record inaccuracy and distribution-free order statistics [18, 19, 23, 24]	Formal protocol-physical risk composition and a finite-sample lower-tail buffer from exchangeable audit errors.
Rights placement	Discrete marginal allocation and compound-distribution recursion [20, 21, 22]	Demand-tail interpretation, component pooling, compound-Poisson implementation, and a no-constant-guarantee result for mean-only placement.
Evaluation	Bootstrap inference and common-random-number variance reduction [25, 26]	Contract-matched models, paired strategy and scenario effects, leader-placement controls, predeclared equivalence margin, and public run-level data.

Table 2: Semantics exposed by the evaluated design families.

Design	Partition response	Budget safety	Possible client-visible loss
Strict R4	Must complete everywhere	Exact	Impossible contract
Fixed-leader CP	May return UNAVAILABLE	Preserved	Unavailability
Escrow ATS Stale AP	Local decision	Preserved Not preserved	False insufficiency Excess commitments or compensation

Proof. R4 implies the three properties contradicted by Theorem 3.1. $\square$

The result does not make inventory systems impossible. It requires an implementation to expose which loss in Table 2 is permitted.

4 Escrow ATS Reservation Primitive

4.1 Object contract and state

Escrow implements a local-rights object, not the global-query object of Definition 2.1 during a partition. Each request identifier x has a designated home $h(x)$.

Definition 4.1 (Escrow ATS). At replica i :

- `LocalAvailable()` $\rightarrow v$ returns rights currently known and spendable by i .
- `LocalReserve( $x, q$ )` $\rightarrow \{\text{OK}, \text{INSUFFICIENT}, \text{INVALID}\}$ executes at $i = h(x)$. A fresh request consumes q local rights when available; otherwise it records `INSUFFICIENT`. A duplicate with the same quantity returns the durable prior outcome, and a quantity mismatch returns `INVALID`.

The operations require no synchronous communication with another replica. A local `INSUFFICIENT` can be false globally because unreachable rights may exist elsewhere.

The convergent accounting state is (R, U) :

- $R \in \mathbb{Z}_{\geq 0}^{k \times k}$ is a non-decreasing transfer matrix. $R[i][j]$ is cumulative rights transferred from i to j .
- $U \in \mathbb{Z}_{\geq 0}^k$ is a non-decreasing consumption vector. $U[i]$ is cumulative successful reservation quantity at i .

Initial allocations a_i^0 satisfy $\sum_i a_i^0 = Q$. Set $R[i][i] = a_i^0$, all off-diagonal entries to zero, and $U[i] = 0$. Diagonal entries are immutable. Only i writes $U[i]$ and off-diagonal entries in row $R[i][\cdot]$. Mutations at i are serialized or use equivalent local concurrency control.

The spendable balance is

$$\text{local}(i) = a_i^0 + \sum_{j \neq i} R[j][i] - \sum_{j \neq i} R[i][j] - U[i]. \quad (1)$$

The home replica also stores a durable single-writer map

$$H_i : \text{RequestId} \rightarrow \mathbb{Z}_{>0} \times \{\text{OK}, \text{INSUFFICIENT}\}.$$

For $i = h(x)$, `LocalReserve( $x, q$ )` is one serialized durable transition:

1. if $H_i[x] = (q, o)$, return o without changing $U[i]$;
2. if $H_i[x] = (q', o)$ with $q' \neq q$, return `INVALID`;
3. if x is unseen and $\text{local}(i) \geq q$, atomically set $U[i] := U[i] + q$ and $H_i[x] := (q, \text{OK})$, then return `OK`;
4. if x is unseen and $\text{local}(i) < q$, durably set $H_i[x] := (q, \text{INSUFFICIENT})$ without changing $U[i]$, then return `INSUFFICIENT`.

A business process that wants to retry after an insufficient outcome must use a new identifier. The successful consumption increment and outcome record must commit atomically before acknowledgment.

A transfer of n from i to j succeeds only if $\text{local}(i) \geq n$, then durably increments $R[i][j]$ before advertising the absolute cumulative value. The source loses the right immediately; the destination spends it only after learning that value.

Accounting states merge by componentwise maximum:

$$(R, U) \sqcup (R', U') = (\max(R, R'), \max(U, U')).$$

A delta must carry absolute inflationary component values; adding a retransmitted delta would duplicate rights. The request map H_i belongs to the durable state of home writer i and is not merged as a numeric CRDT. Replicating or migrating it requires a single-writer or linearizable mechanism.

Proposition 4.1 (Per-request idempotency). *Under a unique home and the atomic transition above, a request identifier increments $U[h(x)]$ at most once. Every same-quantity duplicate returns the first durable outcome.*

Proof. The first durable transition inserts the only entry for x . If it records OK, the same atomic transition increments U once; if it records INSUFFICIENT, it does not increment U . Every later same-quantity invocation takes case 1 and performs no accounting mutation. A different quantity takes case 2. $\square$

4.2 Convergence and conservation

Lemma 4.2 (State convergence). *Replicas that have incorporated the same set of accounting updates hold equivalent (R, U) states. If accounting updates quiesce and every produced update is eventually delivered to every non-failing replica, those replicas eventually hold the same state.*

Proof. The state is ordered componentwise over non-negative integers. Local updates inflate a single-writer component, and merge is the least upper bound. The join is associative, commutative, and idempotent; replicas receiving the same updates therefore hold the same state [10, 9]. After quiescence, eventual delivery gives every non-failing replica the same finite update set. $\square$

For the safety proof, at writer i define cumulative consumption $C_i = U[i]$, outbound transfer $O_i = \sum_{j \neq i} R[i][j]$, and known inbound transfer $I_i = \sum_{j \neq i} R[j][i]$.

Lemma 4.3 (Local conservation invariant). *Every reachable durable writer state at replica i satisfies*

$$C_i + O_i \leq a_i^0 + I_i. \quad (2)$$

Proof. Initially $C_i = O_i = I_i = 0$. A successful reservation increments C_i by q only when $a_i^0 + I_i - O_i - C_i \geq q$, so Equation 2 remains true. An outbound transfer increments O_i under the same precondition. Receiving or merging an inbound transfer can only increase I_i . Single-writer ownership and non-regressing recovery prevent a remote copy from decreasing or independently advancing writer i 's C_i or O_i . Induction over initialization, reservation, transfer, merge, and recovery establishes the invariant. The map H_i changes whether a duplicate can increment C_i , but does not add rights. $\square$

Theorem 4.4 (System no-oversell). *Assume single-writer numeric components, serialized local mutations, atomic durable-before-acknowledgment reservation transitions, durable-before-advertisement transfers, no rollback of completed durable writer*

mutations, and the reservation and transfer preconditions above. In the join of all accounting states,

$$\sum_{i=1}^k U[i] \leq Q.$$

Proof. Let starred quantities denote the join. For each i , C_i^* and O_i^* equal the latest values produced by writer i , while joined inbound information satisfies $I_i^* \geq I_i$ known at that writer. Lemma 4.3 gives $C_i^* + O_i^* \leq a_i^0 + I_i^*$. Summing over i cancels all off-diagonal transfers because $\sum_i O_i^* = \sum_i I_i^*$, leaving

$$\sum_i C_i^* \leq \sum_i a_i^0 = Q.$$

Since $C_i^* = U^*[i]$, the result follows. $\square$

Theorem 4.4 proves quantity conservation. Proposition 4.1 is separately required to refine the idempotent request contract; conservation alone would allow the same business request to consume two legitimate rights under two fresh identifiers or homes.

4.3 Recovery, request retention, and exact queries

A recovered writer may resume only from a state that dominates every completed durable mutation of its own components and request map. For H_i , dominance means retaining every non-garbage-collected request identifier with the same quantity and outcome; recovery may add entries but may not alter an existing result. A system may roll back a transition only if it can prove that the transition was neither acknowledged nor externally observed or advertised. Spending restored known rights is safe without waiting for all anti-entropy traffic; later inbound updates can increase the balance.

If writer state is lost, its remaining rights must be quarantined until an epoch or fencing protocol proves that the old writer cannot reappear. Replica-identity reuse, membership change, right reclamation, and negative inventory adjustment require such a protocol and are not consequences of Theorem 4.4. Home failover must migrate H_i together with writer authority or fence the old home before accepting the same identifiers elsewhere.

Deleting $H_i[x]$ reopens x as fresh. Garbage collection is therefore safe only after a declared retry-retention horizon and a business-level guarantee that the identifier cannot recur, or after a compact tombstone with equivalent uniqueness semantics is retained.

LocalAvailable is exact for the local-rights contract. At an arbitrary partially synchronized replica, $Q - \sum_i U[i]$ can overstate global remaining because remote consumption may be unseen. One sufficient exact GlobalRemaining protocol in a fixed membership epoch is:

1. serialize a barrier at every writer after all prior local reservation transitions;
2. prevent each writer from executing later reservations until the snapshot completes;

3. collect each writer's authoritative $U[i]$ through its barrier and compute $Q - \sum_i U[i]$;
4. release the barriers.

At the last barrier, earlier writers are paused and every included $U[i]$ is authoritative, so the aggregate is exact for that cut. A version-vector or atomic-snapshot protocol can replace the pause if it establishes the same complete frontier. Without such a frontier, the aggregate must be labeled stale or bounded rather than exact.

5 Physical Inventory and Risk Composition

Let $Q_{\text{phys}}, Q_{\text{sys}} \in \mathbb{Z}_{\geq 0}$ be, respectively, pickable physical stock for a reservation cohort and its recorded count at the allocation epoch. Define effective error

$$Q_{\text{phys}} = Q_{\text{sys}} + \epsilon.$$

The distribution of ϵ can include record error and capture delay and can vary by item, location, process, season, and horizon. DeHoratius and Raman found inaccuracies in 65% of nearly 370,000 audited SKU-store records in their dataset [18]; that incidence does not identify a universal error distribution.

Proposition 5.1 (Protocol-physical risk composition). *Let A be accepted reservation quantity. If a protocol guarantees $A \leq Q_{\text{alloc}}$ in every execution and the budgeting rule satisfies*

$$\mathbb{P}(Q_{\text{phys}} < Q_{\text{alloc}} \mid \mathcal{I}) \leq \alpha,$$

then

$$\mathbb{P}(A > Q_{\text{phys}} \mid \mathcal{I}) \leq \alpha.$$

Proof. The event $\{A > Q_{\text{phys}}\}$ is contained in $\{Q_{\text{alloc}} > Q_{\text{phys}}\}$ because $A \leq Q_{\text{alloc}}$. $\square$

With $Q_{\text{alloc}} = \max(0, Q_{\text{sys}} - b)$, a sufficient condition is $\mathbb{P}(\epsilon < -b \mid \mathcal{I}) \leq \alpha$. This is a lower-tail requirement, not a justification for a stationary Gaussian model.

Theorem 5.2 (Distribution-free audit calibration). *Within a fixed operational stratum, let calibration errors $\epsilon_1, \dots, \epsilon_n$ and a future error ϵ_{n+1} be exchangeable. Attach independent $Z_j \sim \text{Uniform}(0, 1)$ and order pairs (ϵ_j, Z_j) lexicographically. Let $r = \lfloor \alpha(n+1) \rfloor$. If $r \geq 1$, let τ be the error component of the r th smallest calibration pair and set $b = \max(0, -\tau)$. Then*

$$\mathbb{P}(\epsilon_{n+1} < -b) \leq \frac{r}{n+1} \leq \alpha.$$

Proof. Because $-b = \min(0, \tau) \leq \tau$, the event $\epsilon_{n+1} < -b$ implies $\epsilon_{n+1} < \tau$. Inserting the future pair among the n calibration pairs then places it among the first r positions. Exchangeability and randomized tie-breaking make its rank uniform on $\{1, \dots, n+1\}$, so the probability is at most $r/(n+1)$. $\square$

Theorem 5.2 is an order-statistic/conformal guarantee [23, 24]. It is marginal over calibration and future error, not arbitrary conditional coverage. A nontrivial r requires $n \geq \lceil 1/\alpha \rceil - 1$: 19 observations at $\alpha = 0.05$, 99 at 0.01, and 999 at 0.001. Stratification improves relevance but reduces sample size; drift breaks exchangeability and must be monitored.

Corollary 5.3. *Under Theorem 4.4 and the buffer in Theorem 5.2, $\mathbb{P}(A > Q_{\text{phys}}) \leq r/(n+1)$ for the calibrated stratum.*

6 Demand-Aware Placement of Rights

6.1 Static no-transfer interval

Let $D_i \in \mathbb{Z}_{\geq 0}$ be requested units at replica i during an interval of length H in which rights cannot cross replica boundaries. Allocate $a_i \in \mathbb{Z}_{\geq 0}$ with $\sum_i a_i = Q_{\text{alloc}}$. The objective is

$$\max_{a_1, \dots, a_k} \sum_{i=1}^k \mathbb{E}[\min(D_i, a_i)] \quad \text{s.t.} \quad \sum_i a_i = Q_{\text{alloc}}. \quad (3)$$

Independence is unnecessary because expectation is separable; only marginal distributions enter this static objective. The objective is unit-granular. For indivisible multi-unit orders, it is an upper-bound relaxation unless partial fulfillment is permitted, because residual rights can be smaller than the next order.

Lemma 6.1 (ATS marginal value). *For integer $a \geq 0$,*

$$\mathbb{E}[\min(D_i, a+1)] - \mathbb{E}[\min(D_i, a)] = \mathbb{P}(D_i \geq a+1).$$

Proof. For every outcome, one additional right fulfills one additional unit exactly when demand is at least $a+1$. $\square$

Corollary 6.2 (Tail-greedy ATS allocation). *Equation 3 is solved by starting at $a_i = 0$ and assigning each successive right to a replica maximizing $\mathbb{P}(D_i \geq a_i + 1)$.*

Proof. Each replica contributes a non-increasing marginal sequence. A feasible allocation selects a prefix from every sequence. Selecting the largest available marginal at each step maximizes the sum; an exchange removes any inversion. This is the ATS specialization of classical discrete marginal allocation [20, 21]. $\square$

For Poisson unit demand, the marginal is a Poisson survival probability. If order arrivals are Poisson and order quantities are independent identically distributed positive integers, requested units follow a compound-Poisson distribution; its exact survival probabilities solve the unit-granular objective above. They do not remove residual-capacity loss when operational reservations are all-or-nothing multi-unit orders. Mean-proportional placement is simpler, but means alone do not determine marginal value.

Proposition 6.3 (No constant guarantee for mean-proportional allocation). *Across a growing number of replicas, allocation proportional only to equal mean demands has no approximation ratio bounded below by a positive constant.*

Proof. Let $k \geq 2$, choose m divisible by k , and set $Q = m$. Let $D_1 = m$ deterministically. For each $i \geq 2$, let $D_i = m^2$ with probability $1/m$ and zero otherwise. Every replica has mean m , so mean-proportional allocation assigns m/k rights to each. Its expected fulfillment is

$$\frac{m}{k} + (k-1) \frac{1}{m} \frac{m}{k} = \frac{m+k-1}{k}.$$

Tail-greedy assigns all m rights to replica 1 because its first m marginals equal one, whereas each bursty marginal equals $1/m$. The optimum is m . The ratio $(m+k-1)/(km) \rightarrow 1/k$ as $m \rightarrow \infty$, which can be made arbitrarily small by increasing k . $\square$

6.2 Known partition components

Corollary 6.4 (Component pooling). *Suppose a fixed partition Π persists for the interval and rights transfer instantaneously within, but not across, each component $C \in \Pi$. Define $D_C = \sum_{i \in C} D_i$ and $a_C = \sum_{i \in C} a_i$. An optimal component budget is obtained by repeatedly maximizing $\mathbb{P}(D_C \geq a_C + 1)$. Any within-component placement is equivalent under the instantaneous-transfer assumption.*

This corollary does not make a policy globally optimal when partitions change, reconnections allow rebalancing, transfers take time, or forecasts are wrong. The dynamic escrow-tail policy evaluated below is a specified heuristic inspired by Corollary 6.2, not a theorem-backed optimum for the full process.

7 Reproducible Evaluation

7.1 Questions and strategy semantics

The experiment asks:

1. How do AP excess commitments, CP unavailability, and escrow fragmentation vary with partition exposure?
2. How strongly does CP depend on rerouting and leader placement?
3. Does the dynamic tail heuristic materially differ from proportional placement in the tested demand models?
4. How do forecast reversal and non-unit quantities affect the conclusions?

Within a scenario and run, all strategies receive identical demand events, quantities, partition intervals, component assignments, and reroute draws.

- **AP-stale:** each partition component spends from the same stale remaining-budget snapshot. This is one invariant-free baseline, not all AP systems.
- **CP-affinity:** one fixed leader serializes reservations; during a partition, only clients in its component reach it.
- **CP-reroute:** an outside client reaches that leader independently when a shared uniform draw is below ρ ; no second leader is introduced.

- **Escrow-proportional:** conserved rights are rebalanced by forecast expected-unit rates at reconnection; transfer inside the current component is instantaneous.
- **Escrow-tail:** the same model uses exact requested-unit demand-tail marginals over the remaining horizon: Poisson for unit orders and compound Poisson for the mixed-quantity scenario. With all-or-nothing multi-unit reservations, this remains a placement heuristic for the unit-granular relaxation.

Connected and partitioned durations alternate as exponential random variables. A partition assigns replicas uniformly to two non-empty components. Order arrivals are independent Poisson processes. The base quantity is one. The mixed sensitivity uses quantities 1, 2, 5 with probabilities 0.80, 0.15, 0.05.

7.2 Common-random-number design

Twelve scenarios are organized into five trace families. Each family has 1,000 independent run seeds. Related scenarios reuse the same seed, with separate deterministic random streams for standard-exponential interval variates, component assignments, demand times, quantities, reroute draws, and random leader selection; this is a common-random-number variance-reduction design [26].

The base, $\rho = 0.50$, and $\rho = 0.95$ five-replica scenarios have identical complete traces; the shared reroute draw makes success monotone in ρ . Correct-forecast, reversed-forecast, high-demand-leader, low-demand-leader, and random-leader variants of the asymmetric case also have identical complete traces. The 25% and 75% partition scenarios share demand and the underlying standardized network variates, while their duration parameters transform those variates into different interval schedules. Cross-scenario effects are therefore paired common-random-number estimates rather than comparisons of unrelated endpoint samples.

7.3 Parameters, metrics, and inference

The base uses $Q = 200$, $T = 3600$ seconds, total order rate 0.06/s, mean connected and partition durations of 300 seconds, and $\rho = 0.8$. Scenarios cover $k = 2, 5, 10$; symmetric and asymmetric demand; reversed allocation forecasts; partition exposure near 25%, 50%, and 75%; $\rho \in \{0.50, 0.80, 0.95\}$; high-, low-, and uniformly random leader placement; and mixed quantities.

For the mixed-quantity scenario, the order rate is 0.045/s. Mean order quantity is 1.35, so the expected requested-unit rate is 0.06075/s, approximately matching the base unit-demand intensity. Compound-Poisson tail probabilities are computed by the Panjer recursion [22] and checked against brute-force allocation on small cases.

Let D be requested units, A accepted units, and

$$B = \min(A, Q).$$

The artifact calls B `budget_feasible_units`; more precisely, it is budget-capped accepted quantity. The reported ratios are

$$E_B = \frac{B}{\min(Q, D)}, \quad E_D = \frac{B}{D},$$

with value one when the respective denominator is zero. E_B is budget-normalized acceptance efficiency relative to the unit upper bound; E_D is demand service ratio. Neither is labeled conventional fill rate. For CP and escrow, $B = A$. For AP-stale, B , E_B , and E_D are upper bounds on quantity or ratios that could be honored after excess commitments; actual compensation and whole-order selection are not modeled. Oversell is $\max(0, A - Q)$. Unavailability and fragmentation are recorded in rejected requested units.

Marginal means use 95% percentile-bootstrap intervals with 2,000 resamples and seed 20260811. Within-scenario strategy differences are paired by run and use seed 20260812. Cross-scenario common-random-number effects use seed 20260813. One bootstrap index matrix is reused across all metrics in a group. Intervals are pointwise; no family-wise multiplicity guarantee is claimed. A tail-minus-proportional 95% interval entirely inside ± 0.5 percentage points of E_B is declared numerically equivalent for this model study. The margin is a predeclared reporting convention, not an empirically calibrated business indifference zone.

7.4 Dynamic results

Table 3 shows the contract distinction. AP-stale reaches the budget-normalized upper bound but accepts 3.25 excess units on average; this is not a fulfillment claim. CP and escrow preserve the budget. Rerouting raises CP E_B from 82.47% to 98.77%; the paired increase is 16.300 points [15.775, 16.842]. Escrow-tail reaches 99.92% E_B and 92.33% E_D .

Table 5 supports directional sensitivity claims with paired effects. Raising partition exposure increases AP oversell and escrow fragmentation while sharply reducing affinity-bound CP efficiency. Raising ρ improves CP-reroute by 4.54 points. Reversing the allocation forecast reduces escrow-tail E_B by 1.46 points and adds 8.60 fragmented units. In the asymmetric trace, placing the leader at the lowest-demand replica reduces CP-affinity by 9.07 points relative to the favorable highest-demand placement; a uniformly random leader reduces it by 5.47 points. Leader location is therefore a material model parameter, not an innocuous implementation detail.

Every tail-minus-proportional E_B interval in all twelve dynamic scenarios lies inside the predeclared ± 0.5 -point margin. This model-level numerical-equivalence result is not a business indifference claim and does not imply that demand-shape information is generally irrelevant. Under reversed forecasts, tail is 0.101 points below proportional and 0.966 points below well-routed CP. At $\rho = 0.95$, the escrow-tail advantage over CP-reroute narrows to 0.127 points.

7.5 Exact allocation benchmark

The static benchmark evaluates Equation 3 exactly from survival functions; it does not use Monte Carlo. Table 7 separates the theorem-aligned static question from the dynamic network heuristic.

The Poisson cases explain why the two dynamic escrow policies are nearly identical under the tested parameters. The

equal-mean stress tests demonstrate Proposition 6.3: equal means can hide radically different marginal values.

7.6 Artifact and threats to validity

The canonical artifact is <https://github.com/peyidev/inventory-cap-simulation> [1]. Its repository root contains the exact simulator, 60,000-row run-level CSV, marginal summary, paired strategy comparisons, paired scenario effects, exact allocation benchmark, generated metadata, pinned dependencies, tests, manuscript source, PDF, and file hashes. `ARTIFACT_IDENTITY.json` declares the repository and validation command; `SHA256SUMS` identifies every archived file; and `make verify` checks repository identity, numerical claims, tests, and integrity. Internal checks compare tail allocation with brute-force enumeration for Poisson and compound-Poisson cases, exercise durable idempotency transitions, verify common traces, and enforce CP/escrow conservation on mixed quantities.

The evaluation remains intentionally stylized. AP-stale is one unsafe baseline. CP omits election, quorum latency, retry traffic, and a concrete consensus implementation. Rerouting is exogenous. Partitions are random bipartitions with exponential durations rather than measured incidents. Demand is independent Poisson or compound Poisson and does not represent correlated flash demand. The high-demand leader is a favorable CP configuration, now bracketed by low-demand and random controls. Escrow transfer inside a component is instantaneous and free. The model excludes cancellations, expiry, receipts, negative adjustments, multi-SKU atomic reservations, adversarial schedules, and physical-stock error. The physical calibration theorem is not empirically instantiated because no audit dataset is supplied. Numerical conclusions therefore apply only to the specified contracts, stochastic processes, and common-random-number transformations.

8 Related Work

Gilbert and Lynch formalized the impossibility of atomic consistency and availability under partition [2]. Brewer emphasized a continuum of design choices, and PACELC separates partition-time trade-offs from normal-operation latency and consistency [3, 4]. Theorem 3.1 specializes the issue to an ATS commitment budget and states the independent operational properties directly.

O’Neil introduced escrow transactions for distributed aggregate constraints [6]. Preguiça et al. used leased reservations to guarantee disconnected mobile transactions, including stock-like escrow reservations and workload-based allocation experiments [7]. Exo-leasing applied escrow synchronization to intermittently connected commodity-storage clients [8]. Balegas et al. represented bounded counters as CRDTs with preallocated rights [9]. This paper does not claim those mechanisms as new; it adds an ATS-specific contract, atomic request-idempotency state, recovery and exact-query boundaries, physical-risk composition, and a synchronized artifact.

Table 3: Base case: $k = 2$, symmetric demand, mean partition fraction 47.8%. Values are means with 95% percentile-bootstrap intervals. B is budget-capped accepted quantity; E_B and E_D are percentages.

Strategy	Accepted units	Oversell units	B units	E_B	E_D
AP-stale	202.38 [201.95, 202.84]	3.25 [2.89, 3.64]	199.13 [198.93, 199.34]	100.00 [100.00, 100.00]	92.41 [92.08, 92.75]
CP-affinity	164.28 [163.07, 165.49]	0.00 [0.00, 0.00]	164.28 [163.07, 165.49]	82.47 [81.88, 83.04]	75.95 [75.49, 76.39]
CP-reroute	196.71 [196.32, 197.10]	0.00 [0.00, 0.00]	196.71 [196.32, 197.10]	98.77 [98.64, 98.90]	91.20 [90.93, 91.44]
Escrow-prop.	198.96 [198.74, 199.16]	0.00 [0.00, 0.00]	198.96 [198.74, 199.16]	99.92 [99.89, 99.94]	92.33 [92.00, 92.64]
Escrow-tail	198.96 [198.75, 199.16]	0.00 [0.00, 0.00]	198.96 [198.75, 199.16]	99.92 [99.89, 99.94]	92.33 [92.01, 92.65]

Table 4: Sensitivity means. Efficiency and service columns are percentages. “Frag.” is rejected requested quantity while sufficient rights still exist globally but not in the requester’s component.

Scenario	Part.	AP over.	CP-aff. E_B	CP-route E_B	Escrow E_B	Escrow E_D	Frag.
$k = 5$, symmetric	48.6	2.67	85.53	98.91	99.95	92.58	1.59
$k = 5$, asymmetric, high-demand leader	47.3	2.54	92.02	99.43	99.93	92.06	1.94
$k = 5$, reversed forecast	47.3	2.54	92.02	99.43	98.47	90.66	10.54
$k = 5$, low-demand leader	47.3	2.54	82.95	98.87	99.93	92.06	1.94
$k = 5$, random leader	47.3	2.54	86.55	99.12	99.93	92.06	1.94
$k = 5$, partition 25%	24.9	0.80	94.78	99.54	99.98	92.62	0.73
$k = 5$, partition 75%	73.2	5.13	74.61	98.03	99.90	92.54	2.71
$k = 5$, $\rho = 0.50$	48.6	2.67	85.53	95.28	99.95	92.58	1.59
$k = 5$, $\rho = 0.95$	48.6	2.67	85.53	99.82	99.95	92.58	1.59
$k = 5$, mixed quantities	47.5	2.69	86.95	98.80	99.86	91.19	2.22
$k = 10$, symmetric	48.0	2.90	84.86	98.96	99.95	92.41	1.66

Table 5: Paired cross-scenario effects with 95% paired-bootstrap intervals. Efficiency effects are percentage points; other effects are units. The value is left scenario minus right scenario.

Effect	Mean [95% CI]
Partition 75 minus 25: AP oversell	+4.33 [+3.93, +4.76]
Partition 75 minus 25: CP-affinity E_B	-20.18 [-20.64, -19.68]
Partition 75 minus 25: escrow fragmentation	+1.99 [+1.74, +2.24]
$\rho = 0.95$ minus 0.50: CP-reroute E_B	+4.54 [+4.25, +4.85]
Reversed minus correct forecast: escrow E_B	-1.46 [-1.68, -1.25]
Reversed minus correct forecast: fragmentation	+8.60 [+7.92, +9.27]
Low- minus high-demand leader: CP-affinity E_B	-9.07 [-9.64, -8.50]
Random minus high-demand leader: CP-affinity E_B	-5.47 [-5.97, -4.99]

Coordination avoidance characterizes when invariants can be preserved without coordination [11], while CALM relates coordination freedom to monotonicity [12]. The Homeostasis protocol derives workload-optimized treaties for independent execution [13]; predictive treaties add time-dependent forecasts to reduce coordination while preserving consistency [14]. Just-Right Consistency and IPA develop principled invariant-preserving application designs under weak consistency [15, 16]. MRVs split numeric values across records to reduce transactional hotspot conflicts while retaining the underlying isolation model [17]; that objective differs from partition-time local availability. The present scope is narrower: one finite ATS budget, explicit client-visible contract losses, and demand-aware placement of delegated rights.

Fox’s marginal analysis and Federgruen and Groenevelt’s greedy conditions establish the resource-allocation foundation used in Corollary 6.2 [20, 21]. The survival-probability marginal is specific to expected ATS units, while Proposition 6.3 isolates a failure mode of mean-only placement.

Inventory record inaccuracy is empirically consequential [18, 19]. Wilks’ tolerance limits and conformal prediction provide distribution-free finite-sample coverage under exchange-

ability [23, 24]. Theorem 5.2 maps that guarantee to an ATS lower-tail buffer; it does not solve drift or arbitrary conditional coverage.

Sagas and compensation can repair a workflow after a commitment fails [27]. Compensation does not prevent the initial commitment total from exceeding a shared invariant.

9 Data and Code Availability

The complete artifact is maintained at <https://github.com/peyidev/inventory-cap-simulation> [1]. A repository checkout is validated with `make verify`. Artifact version 3.1.0 together with the root SHA256SUMS manifest identifies the exact files underlying the reported results; `ARTIFACT_IDENTITY.json` records the canonical repository and expected filenames.

10 Operational Implications

The formal results yield concrete design rules:

- Use fixed-leader or quorum-backed CP when exact global ATS dominates and routing can tolerate or mask leader unreachability; evaluate leader placement explicitly.
- Use escrow when non-blocking local reservation and a hard system budget dominate, while accepting false insufficiency and operating a rights-rebalancing control loop.
- Commit consumption and the request outcome atomically before acknowledging; retain or tombstone request identifiers through the retry horizon.
- Fence lost writer and home epochs before reclaiming rights or reusing authority.
- Do not expose a partially synchronized aggregate as exact global remaining inventory; establish an epoch-complete snapshot frontier.

Table 6: Paired within-scenario mean differences with 95% intervals. Efficiency differences are percentage points. Columns are tail minus proportional, tail minus CP-reroute, CP-reroute minus CP-affinity, and AP minus escrow-tail oversell.

Scenario	Tail-prop. E_B	Tail-CP E_B	Route-aff. E_B	AP oversell
$k = 2$, base	+0.000 [+0.000, +0.000]	+1.148 [+1.016, +1.280]	+16.300 [+15.775, +16.842]	+3.25 [+2.92, +3.62]
$k = 5$, asymmetric	+0.001 [-0.005, +0.008]	+0.499 [+0.421, +0.583]	+7.418 [+7.044, +7.828]	+2.54 [+2.27, +2.82]
Reversed forecast	-0.101 [-0.120, -0.082]	-0.966 [-1.182, -0.769]	+7.418 [+7.013, +7.822]	+2.54 [+2.27, +2.82]
Partition 75%	+0.000 [+0.000, +0.000]	+1.868 [+1.699, +2.038]	+23.420 [+22.836, +24.004]	+5.13 [+4.71, +5.59]
$\rho = 0.50$	+0.000 [+0.000, +0.000]	+4.668 [+4.342, +4.982]	+9.744 [+9.381, +10.100]	+2.67 [+2.37, +2.98]
$\rho = 0.95$	+0.000 [+0.000, +0.000]	+0.127 [+0.096, +0.160]	+14.285 [+13.670, +14.896]	+2.67 [+2.39, +2.98]
Mixed quantities	+0.000 [+0.000, +0.000]	+1.058 [+0.943, +1.193]	+11.858 [+11.288, +12.415]	+2.69 [+2.39, +3.03]

Table 7: Exact expected fulfillment for mean-proportional and tail-optimal allocations. The bursty cases are constructed stress tests, not fitted production demand.

Case	Proportional allocation	Tail allocation	Prop. $\mathbb{E}[F]$	Tail $\mathbb{E}[F]$	Prop./opt.
Poisson, skewed means, $Q = 30$	[18, 7, 3, 1, 1]	[21, 7, 2, 0, 0]	29.23	29.79	0.981
Equal-mean bursty, $k = 5$, $Q = 100$	[20, 20, 20, 20, 20]	[100, 0, 0, 0, 0]	20.80	100.00	0.208
Equal-mean mixed shapes, $k = 3$, $Q = 30$	[10, 10, 10]	[20, 10, 0]	21.99	29.99	0.733
Symmetric Poisson, $k = 5$, $Q = 50$	[10, 10, 10, 10, 10]	[10, 10, 10, 10, 10]	49.96	49.96	1.000

- Calibrate Q_{alloc} below recorded stock from audited lower-tail error rather than treating protocol no-oversell as physical accuracy.
- Monitor fragmentation, forecast calibration, partition exposure, leader reachability, transfer delay, capture lag, and cancellation rate as separate service indicators.

A production inventory service generally combines CP for membership and epoch changes, escrow for a bounded reservation fast path, event capture for stock movement, and an audit-calibrated buffer for physical uncertainty.

11 Conclusion

A strict real-time-inventory requirement that combines the sequential ATS object with completion at every non-failing replica is impossible through arbitrary partitions. More generally, a system cannot simultaneously guarantee a global commitment budget, local termination, and no false rejection in quiescent histories. A viable system must expose a weaker contract.

Escrow preserves a hard reservation budget without synchronous coordination on each sale, but it exposes local-rights semantics. Full request correctness additionally requires an atomic durable outcome map, stable home authority, non-regressing recovery, request-retention rules, and fenced reconfiguration. Exact global remaining requires a complete snapshot frontier. None of these system properties implies physical-stock accuracy; a separately calibrated allocation budget is required.

Rights placement is a resource-allocation problem. Demand-tail marginals exactly solve the static unit-granular no-transfer objective, including compound-Poisson requested-unit demand; all-or-nothing multi-unit orders introduce residual-capacity effects outside that theorem. Mean-only placement has no general approximation guarantee. In the declared dynamic models, proportional and tail policies fall within the predeclared numerical-equivalence margin, but forecast reversal materially harms escrow. CP performance changes with rerouting and leader location. Correctness and performance claims

are therefore meaningful only after the reservation contract, routing and leader assumptions, failure process, demand model, and physical-risk budget are stated separately.

References

- [1] P. A. Laris Uribe. *Available-to-Sell Reservations Under Network Partitions: Reproducibility Artifact*, version 3.1.0. GitHub, 2026. <https://github.com/peyidev/inventory-cap-simulation>.
- [2] S. Gilbert and N. Lynch. Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2):51–59, 2002.
- [3] E. A. Brewer. CAP twelve years later: How the “rules” have changed. *Computer*, 45(2):23–29, 2012.
- [4] D. J. Abadi. Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. *Computer*, 45(2):37–42, 2012.
- [5] M. P. Herlihy and J. M. Wing. Linearizability: A correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems*, 12(3):463–492, 1990.
- [6] P. E. O’Neil. The escrow transactional method. *ACM Transactions on Database Systems*, 11(4):405–430, 1986.
- [7] N. Preguiça, J. Legatheaux Martins, M. Cunha, and H. Domingos. Reservations for conflict avoidance in a mobile database system. In *First International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*. USENIX Association, 2003.
- [8] L. Shriram, H. Tian, and D. Terry. Exo-leasing: Escrow synchronization for mobile clients of commodity storage servers. In *Proceedings of the ACM/IFIP/USENIX Middleware Conference*, pages 42–61, 2008.
- [9] V. Balesgas, D. Serra, S. Duarte, C. Ferreira, M. Shapiro, R. Rodrigues, and N. Preguiça. Extending eventually consistent cloud databases for enforcing numeric invariants. In *Proceedings of the IEEE Symposium on Reliable Distributed Systems*, pages 31–36, 2015.
- [10] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski. A comprehensive study of convergent and commutative replicated data types. Technical Report RR-7506, INRIA, 2011.
- [11] P. Bailis, A. Fekete, M. J. Franklin, A. Ghodsi, J. M. Hellerstein, and I. Stoica. Coordination avoidance in database systems. *Proceedings of the VLDB Endowment*, 8(3):185–196, 2014.
- [12] J. M. Hellerstein and P. Alvaro. Keeping CALM: When distributed consistency is easy. *Communications of the ACM*, 63(9):72–81, 2020.
- [13] S. Roy, L. Kot, G. Bender, B. Ding, H. Hojjat, C. Koch, N. Foster, and J. Gehrke. The Homeostasis protocol: Avoiding transaction coordination through program analysis. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1311–1326, 2015.

- [14] T. Magrino, J. Liu, N. Foster, J. Gehrke, and A. C. Myers. Efficient, consistent distributed computation with predictive treaties. In *Proceedings of the Fourteenth EuroSys Conference*, article 36, pages 1–16, 2019.
- [15] M. Shapiro, A. Bieniusa, N. Preguiça, V. Balegas, and C. Meiklejohn. Just-Right Consistency: Reconciling availability and safety. *CoRR*, abs/1801.06340, 2018.
- [16] V. Balegas, S. Duarte, C. Ferreira, R. Rodrigues, and N. Preguiça. IPA: Invariant-preserving applications for weakly consistent replicated databases. *Proceedings of the VLDB Endowment*, 12(4):404–418, 2018.
- [17] N. Faria and J. Pereira. MRVs: Enforcing numeric invariants in parallel updates to hotspots with randomized splitting. *Proceedings of the ACM on Management of Data*, 1(1), article 43, pages 1–27, 2023.
- [18] N. DeHoratius and A. Raman. Inventory record inaccuracy: An empirical analysis. *Management Science*, 54(4):627–641, 2008.
- [19] A. Shabani, G. Maroti, S. de Leeuw, and W. Dullaert. Inventory record inaccuracy and store-level performance. *International Journal of Production Economics*, 235:108111, 2021.
- [20] B. Fox. Discrete optimization via marginal analysis. *Management Science*, 13(3):210–216, 1966.
- [21] A. Federgruen and H. Groenevelt. The greedy procedure for resource allocation problems: Necessary and sufficient conditions for optimality. *Operations Research*, 34(6):909–918, 1986.
- [22] H. H. Panjer. Recursive evaluation of a family of compound distributions. *ASTIN Bulletin*, 12(1):22–26, 1981.
- [23] S. S. Wilks. Determination of sample sizes for setting tolerance limits. *The Annals of Mathematical Statistics*, 12(1):91–96, 1941.
- [24] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.
- [25] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.
- [26] A. M. Law. *Simulation Modeling and Analysis*. McGraw-Hill Education, 5th edition, 2015.
- [27] H. Garcia-Molina and K. Salem. Sagas. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 249–259, 1987.