



One Object, One State, One Location: The Transition Ledger at the Centre of a Warehouse Management System

Architecture, Computational Methods, and Field Evidence from MileSoft Warehouse Management

MileSoft Engineering Research Group

MileSoft Software Technologies

<https://milessoft.net/>

Corresponding authors: *Deepak Daryani* (deepakdaryani@milessoft.net), *Tanuj Daryani* (tanujdaryani@milessoft.net)

Working paper — Version 1.0 — 24 August 2026

Permanent link: <https://milessoft.net/research/products/wms>

Abstract

A warehouse management system is usually sold as a collection of workflows — receiving, putaway, picking, packing, shipping, counting. That framing obscures what the software actually is. Underneath every one of those workflows is a single ledger asserting, for each physical object, exactly one current state and exactly one current location, and every workflow is a guarded transition on that pair.

This paper presents the architecture and computational methods of MileSoft Warehouse Management from that position. We state the two-coordinate invariant and the four obligations a transition must satisfy to preserve it, and show that the master data layer — item, location, vendor and user records — is not administrative overhead but the precondition that makes a transition validatable at all. A location carries a type and a capacity, so a putaway is feasible only where the unit's state and the location's type agree and the capacity is not exceeded.

Three computational results follow. Directed putaway is a constrained assignment problem whose objective is travel-weighted velocity slotting, and permitting an ad-hoc override is a deliberate trade of slotting quality for receiving throughput that can be priced rather than merely allowed. Exception-triggered cycle counting is characterised by a coverage rate and an expected record staleness, which is the quantity a continuous count controls and a periodic count does not. And integration effort scales with the number of client mappings against a stable contract rather than with the number of custom pipelines, which is the mechanism behind the published onboarding result.

A multi-client third-party logistics deployment reported new-client onboarding falling from six weeks to eight days, and that figure is attributed in its own source to this module's interface architecture. The same deployment's inventory-accuracy and dock-idle improvements are attributed to other modules and are not claimed here.

Keywords: Warehouse management system; WMS software; Directed putaway; Slotting optimization; Cycle counting; Inventory accuracy; Multi-client 3PL warehouse; REST API warehouse integration; Zone rack bin location master; Wave planning and picking; Barcode and RFID warehouse; Distribution centre operations

1 Introduction

Every distribution-centre operator eventually meets the ceiling. A system that was adequate for one client and eighty thousand square feet begins to crack at three clients and two hundred thousand. Onboarding takes six weeks. Custom workflows accumulate. The day the operations team starts saying they will do something in a spreadsheet until the software can be changed, the warehouse management system has become the bottleneck it was bought to remove.

That is the product's own diagnosis and it is a good one. What this paper adds is a way to say precisely what such a system is, so that a buyer can test whether a candidate is the calm spine of a facility or the next ceiling.

The definition offered here is narrow on purpose. A warehouse management system is a ledger asserting, for each physical object, one current state and one current location. Everything else it does — receiving, putaway, wave planning, picking, packing, dispatch, counting — is a guarded transition on that pair. Sections 3 and 4 develop the consequences.

1.1 Why the workflow framing misleads

Software in this category is compared feature by feature: does it do wave picking, does it do cluster picking, does it handle hazmat. Those comparisons rarely predict which system a facility will end up routing around, because every serious candidate has the features.

What separates them is whether the ledger stays true. A system permitting a movement it cannot record, or recording a movement that did not happen, degrades until its numbers are advisory — and a facility with advisory numbers reverts to physical verification, which is what the spreadsheet actually is.

A warehouse routes around its system when the system's answer stops being worth checking. That is a property of the invariant, not of the feature list.

1.2 Contributions

1. The two-coordinate invariant and the four obligations a transition must satisfy to preserve it (Sections 3.1 and 4.1).
2. Master data stated as the precondition for validation, with location type and capacity as hard feasibility constraints on putaway (Sections 3.2 and 4.2).
3. Directed putaway as a constrained assignment problem, and the cost of an ad-hoc override expressed in the objective it degrades (Section 4.2).

4. Exception-triggered cycle counting characterised by coverage rate and expected record staleness (Section 4.3).
5. Integration effort as mappings against a stable contract rather than per-client pipelines, which is the mechanism behind the published onboarding result (Sections 4.4 and 5.2).
6. An eight-dimension capability reference framework for warehouse management systems (Section 6.3).

2 Background and Related Work

Warehouse operations have an unusually good quantitative literature and an unusually poor connection between it and the software that runs them. The results below are settled; what is rarely stated is which of them a given system is actually implementing.

2.1 Travel is the cost, and slotting is the lever

De Koster, Le-Duc and Roodbergen's review of order-picking establishes the finding every design in this field rests on: travel dominates picking time. Picking itself is fast; walking between pick faces is not.

Two families of response follow. Routing policies decide the sequence in which a picker visits faces — Roodbergen and de Koster on aisle routing, Petersen's comparison of routing policies in a conventional warehouse — and slotting policies decide which face a unit occupies in the first place. Slotting is the more durable lever because it changes the distance every subsequent pick must travel.

Gu, Goetschalckx and McGinnis survey the design decisions this implies at facility level. Their contribution for present purposes is the ordering: layout constrains slotting, slotting constrains routing, and a routing improvement inside a bad slotting is bounded by the slotting.

2.2 Class-based storage and where its assumptions bind

Class-based storage descends from Dickie's value classification but substitutes movement for value: fast movers occupy faces near the depot, slow movers occupy the back. Its efficiency gain over random storage is well established and its assumption is easy to miss — it requires demand to be stable enough that yesterday's velocity predicts tomorrow's travel.

This is why the product's material treats slotting analytics as continuous rather than annual. A class assignment computed once a year is a prediction with a one-year horizon, and the literature's efficiency results assume a horizon over which the classification holds.

Continuous re-slotting is not a convenience feature. It is what keeps the assumption behind class-based storage true for long enough to deliver the result the literature promises.

2.3 Identity, events and the boundary above

A transition ledger needs identifiers it can bind to. GS1 supplies the Global Trade Item Number for a trade item class and the Serial Shipping Container Code for an individual logistic unit; ISO/IEC 15459-1 covers transport units outside that scheme; the General Specifications govern symbology and print quality, which decide whether a scan succeeds at the first attempt on a cold aisle.

The event model is EPCIS, standardised as ISO/IEC 19987, with its vocabulary in ISO/IEC 19988. A transition is an event with a business step and a disposition, which is what allows a warehouse ledger to be a traceability record rather than a private log.

IEC 62264-3 places inventory operations as a defined activity with a specified interface to the enterprise system of record. The practical consequence, and the one Section 4.4 develops, is that the interface is the product boundary: a system whose workflows are reachable only through its own screens has no boundary, and every integration becomes bespoke.

3 System Overview

The system is organised as a governed master-data layer, a transition ledger over it, and a set of task generators — putaway, wave, pick, dispatch, count — that propose transitions the ledger then guards. The captured deployment presents these as tabs: master data, identifier management, inbound, planning, picking, dispatch, reports and traceability.

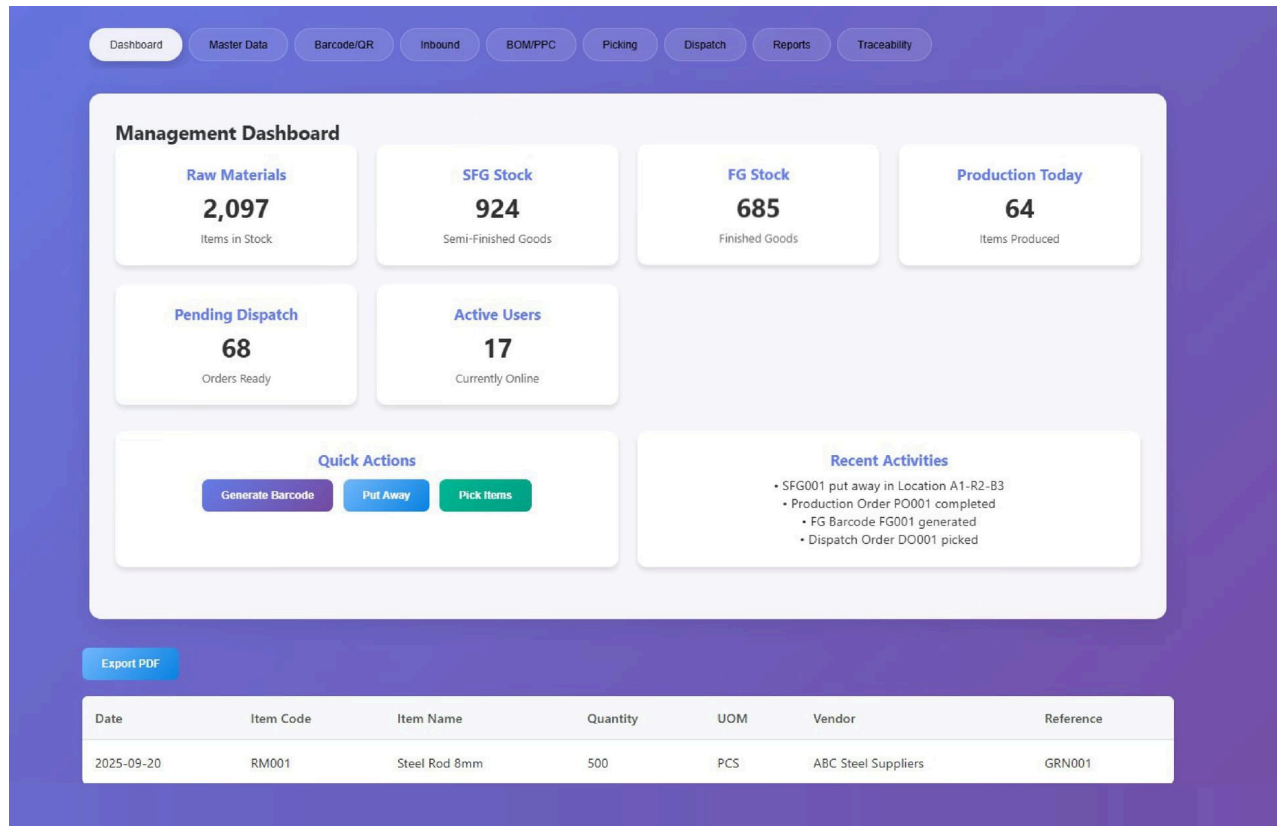


Figure 1. Product capture: the management dashboard. Stock is reported as three separate positions — raw material, semi-finished and finished goods — beside production completed today, orders ready for dispatch and users currently online, with quick actions for identifier generation, putaway and picking, and a recent-activity list in which each entry names a transition and the location it touched.

3.1 The two coordinates

Every tracked object carries a state and a location. In the captured build the states are raw material, semi-finished and finished goods, and the location is a code composed from zone, rack and bin.

It is worth being explicit that this build spans production. A pure distribution warehouse has one stock state and many locations; this one has three states, a production-planning tab and a traceability tab, because material is transformed inside the facility rather than only moved through it. The state coordinate is doing real work here rather than being a formality.

Table 1. The transitions the system generates, and what each changes. Every row alters exactly one coordinate or both, and none alters neither.

Transition	State	Location
Goods receipt	unset to raw material	unset to receiving area
Putaway	unchanged	receiving area to a typed storage location
Issue to production	unchanged	storage to line side
Production completion	raw to semi-finished, or semi-finished to finished	line side to a location of the matching type
Pick	unchanged	storage to staging
Dispatch	finished goods to shipped	staging to outbound
Cycle count	unchanged	unchanged — it writes an observation, and an adjustment only on variance

The last row is what distinguishes a ledger from a database. A count is an observation about the ledger, not an instruction to it; the adjustment it may cause is a separate, attributable transition.

3.2 Four masters, and what each makes checkable

Master data is presented in the captured build as four registers — users, locations, items and vendors — and each exists to make a different part of a transition validatable.

Table 2. The four master registers and the transition obligation each supports.

Register	What it holds	What it makes checkable
Item master	item type, unit of measure, minimum stock, current stock	That the thing being moved is a thing the system knows, in units it can add up
Location master	zone, rack, bin, location type, capacity, current stock	That the destination exists, accepts this state, and has room
Vendor master	supplier records with contact details and status	That a receiving discrepancy can be attributed rather than merely counted
User master	role-based accounts by employee and department	That the person performing the movement is permitted to perform it

The location register carries the most weight in Section 4.2, because it holds two fields that turn putaway from advice into arithmetic: a type, so a semi-finished unit cannot be stored where the facility has designated something else, and a capacity against current stock, so a suggested destination can be full.

The screenshot shows a web application interface for Master Data Management. At the top, there is a navigation bar with tabs: Dashboard, Master Data (selected), Barcode/QR, Inbound, BOM/PPC, Picking, Dispatch, Reports, and Traceability. Below the navigation bar, the 'Master Data Management' section is active, with sub-tabs for Users, Locations (selected), Items, and Vendors. The 'Location Master' form contains input fields for Zone (e.g., A, B, C), Rack (e.g., R1, R2, R3), Bin (e.g., B1, B2, B3), and a Location Type dropdown (Select Type). A 'Save Location' button is below the form. Below the form is a table with columns: Location Code, Zone, Rack, Bin, Type, Capacity, Current Stock, and Actions. The table contains one row with data: A1-R2-B3, A1, R2, B3, SFG Store, 100, 45. The Actions column for this row has 'Edit' and 'Generate Barcode' buttons. Below the table is an 'Export PDF' button. At the bottom of the interface is a table with columns: Date, Item Code, Item Name, Quantity, UOM, Vendor, and Reference. It contains one row of data: 2025-09-20, RM001, Steel Rod 8mm, 500, PCS, ABC Steel Suppliers, GRN001.

Figure 2. Product capture: the location master. A location is composed from zone, rack and bin into a single code, carries a type and a capacity against its current stock, and can generate its own barcode — which is what allows a putaway to be confirmed against the shelf rather than against a screen.

3.3 Identifiers for both coordinates

The system generates and prints identifiers for items and for locations. That both coordinates are independently scannable is the enabling condition for two-key confirmation: an operator scanning a unit and a shelf asserts two physical facts about each other rather than confirming a screen.

The consequence for print quality is not decorative. Every guard in Section 4 assumes a scan succeeds first time; a label that has to be typed converts a validated transition into an asserted one, and the ledger's guarantees degrade to whatever the operator entered.

4 Computational Methods

Four computations carry the system. The first is the invariant everything else must preserve; the other three are the decisions the system makes on top of it.

4.1 The invariant and its four obligations

Let U be the set of tracked units, S the set of states and L the set of locations, with σ and λ the functions assigning a state and a location to each unit.

every unit has exactly one state and exactly one location, and the volume assigned to any location must not exceed that location's capacity (invariant)

$v(u)$ is the unit's volume in the location's capacity units and $\kappa(l)$ the capacity carried on the location record. The second condition is what makes the location master an operational constraint rather than a description.

Four obligations preserve this. Every physical movement writes exactly one transition; no transition is written without a movement; a transition is rejected unless it resolves against governed masters; and a transition that would violate the capacity or type condition is refused rather than recorded with a warning.

Table 3. The failure modes of the invariant, what each looks like on the floor, and what a facility does in response. All four are common and only the first is usually noticed.

Violated obligation	What it looks like	Facility response
Movement without transition	Picker cannot find stock the system shows	Physical search, then an adjustment nobody can attribute
Transition without movement	System shows stock in a location that is empty	Same search, same adjustment, opposite sign
Ungoverned resolution	Free-text location or item on a record	The record cannot be validated or aggregated; it is excluded from reports
Guard downgraded to warning	Over-capacity or type-mismatched location accepted	Stock stored where nobody looks for it; found at the next count

The first two produce compensating errors that a net reconciliation cancels. A facility measuring only net variance can hold both faults indefinitely and conclude its records are sound.

4.2 Directed putaway, and the price of an override

Directed putaway assigns an incoming unit to a location. The product's material names the rule inputs — velocity, size, weight, hazard class, refrigeration — and the literature of Section 2.1 supplies the objective they are serving.

minimise the sum over units and locations of pick frequency times travel distance, subject to: each unit assigned exactly once, no location over capacity, and no assignment to a location whose type the unit's state does not permit (putaway)

$f(u)$ is the unit's expected pick frequency, $d(l)$ the travel distance from the depot to location l , and $\tau(u,l)$ is 1 where the location type accepts the unit's state and 0 otherwise. The objective is travel-weighted velocity slotting: fast movers near, slow movers far.

Ad-hoc putaway relaxes this. The operator is offered the system's suggestion and may confirm it or select an approved alternative, which the product's material recommends during receiving surges. The trade is real and worth stating as one: throughput now against travel cost on every subsequent pick of that unit.

the cost of overriding is pick frequency times the extra distance, summed over the overridden units; mean cost per override is that total divided by their number (override)

O is the set of overridden putaways in a period. The quantity is measurable from data the system already holds, because both the suggested and the chosen location are recorded.

Two things follow. Overrides on fast movers cost far more than overrides on slow movers, because $f(u)$ multiplies the penalty — so a surge policy that permits override on any unit is worse than one that permits it only below a velocity threshold. And because the system records what it suggested, override cost is reportable, which converts an operational habit into a number a manager can act on.

Ask a candidate system whether it stores the location it suggested alongside the location that was used. Without both, the cost of every override is invisible and the slotting engine cannot be evaluated at all.

4.3 Cycle counting, characterised by staleness

The product's material states that counts run continuously alongside picking, triggered by class, by time since last count and by picker discrepancy, without an operational freeze. What that controls is not the count rate but how long a wrong record survives.

coverage rate $c = \text{counts performed} / (\text{locations} \times \text{periods})$; expected staleness under scheduled counting is half the count interval; under exception triggering it is shortened by the discrepancy detection rate μ (staleness)

Theta is the time between a record becoming wrong and the error being found. μ is the rate at which a picker encountering an unexpected quantity raises a count task — a second, demand-driven discovery channel that a scheduled count does not have.

The μ term is where the design earns its keep, and it has a property worth naming: it is concentrated exactly where errors matter. A location nobody picks from contributes nothing to μ , and an error there costs nothing until it is counted. A fast-moving face is visited many times a period, so an error in it is found almost immediately.

This also explains why an annual freeze count cannot control staleness however thoroughly it is performed. Its coverage rate is one per location per year, so expected staleness is six months, and the errors it finds are on average half a year old — long past the point where they could be attributed to a cause.

Record accuracy — the share of locations whose count matches — is then the operational figure to manage, because it is the probability that any given pick finds what the system promised. The published deployment moved it from 91% to 99.4%, and that result is attributed in its own source to the tracking module rather than to this one; Section 5.3 says so.

4.4 Why an interface changes the shape of onboarding

The published onboarding result — six weeks to eight days for a new client — is the strongest claim attached to this module, and its mechanism is simple enough to state as arithmetic.

with per-client pipelines, effort is the sum of a fixed cost plus a bespoke build for each client; against a stable contract, it is one fixed cost plus a mapping per client (onboard)

a is the fixed cost of the integration surface itself, $b(k)$ the bespoke build for client k , and $m(k)$ the mapping of client k 's fields onto an existing contract. The claim is not that mapping is trivial but that it is bounded, because the target of the mapping does not move.

The distinction that matters is stability rather than protocol. An interface exposed over a well-known transport but changing shape per client offers no reduction: each client still faces a bespoke build, and the fixed cost has merely been paid twice.

The second half of the product's claim is configuration rather than integration — pick paths, putaway rules and packing flows configured rather than coded. That is the same argument applied inside the facility: a workflow change that requires a release is a bespoke build with a longer queue in front of it.

The test is not whether an interface exists. It is whether the same endpoint served the last three clients unchanged.

5 Reported Outcomes and Field Evidence

This module is unusual in this library in having one deployment result attributed to it specifically. That makes the separation between what belongs to it and what belongs to the programme around it more important, not less.

5.1 Observed properties of the software

Table 4. Properties read directly from the delivered software, with where each was observed.

Property	Value	Where observed
Stock states tracked separately	raw material, semi-finished, finished goods	Management dashboard
Location code composition	zone, rack and bin into one code	Location master
Location typing	a location carries a type, e.g. a semi-finished store	Location master
Location capacity	capacity against current stock, per location	Location master
Identifiers for both coordinates	barcode generation for items and for locations	Location master and identifier screens
Master registers	users, locations, items, vendors	Master data screen
Activity record granularity	each entry names a transition and the location it touched	Management dashboard

One observation belongs here rather than in a footnote. The captured build is a manufacturing warehouse — three stock states, a production-planning tab, production orders completing on the activity list — while the product's written material describes a multi-client third-party logistics distribution centre. Both are true of the product; only the first is visible in these captures, and nothing about multi-client operation is demonstrated by them.

5.2 The result attributed to this module

A multi-client third-party logistics operator deployed six MileSoft warehouse modules across a distribution centre. Its previous warehouse system had no interface, so every client integration required roughly six weeks of custom data plumbing. After deployment, new-client onboarding is reported at eight days.

The published account attributes this specifically and mechanically: client integrations became interface connections with standard mappings rather than custom-coded pipelines. That is the mechanism Equation (onboard) states, and it is the reason this figure is reported here as belonging to this module while the deployment's other results are not.

Attribution, stated plainly. The onboarding result is this module's. The accuracy improvement is attributed in the same source to the tracking module, and the dock-idle reduction to the dock and yard modules. None of those is claimed here.

Two caveats belong with the figure even so. It is a single operator's experience with no sample of clients stated, and the six-week baseline is a property of the system replaced rather than of the category — an operator whose previous system already had a stable interface should expect a much smaller change.

5.3 The results that belong to the programme

Table 5. The deployment's other reported outcomes and the module each is attributed to in its own source. None is claimed for this module.

Outcome	Value	Attributed to
Inventory accuracy	91% to 99.4%	Inventory Tracking — handheld and RFID cycle counting
Dock idle time	38% reduction	Loading Dock Management with Truck Turnaround
Average truck dwell	3 h 10 min to 1 h 20 min	Truck Turnaround with Loading Dock Management

The accuracy figure deserves a note because it is the one most easily misattributed. Section 4.3 explains why continuous counting controls staleness, and this module generates the tasks; but the published account credits the handheld and radio-frequency capture that made counting possible alongside picking. Both statements are consistent, and the honest reading is that the result required the pair.

5.4 Modelled labour recovery

The published return model for this module is stated as labour recovered across receiving, picking, packing and shipping. Its assumptions are printed here so a reader can substitute their own.

Table 6. Modelled scenario — not a deployment result. Assumptions are the published defaults of the return-on-investment model for this product.

Assumption	Value
Warehouse staff in scope	15
Labour gain attributed to the module	15%
Modelled recovery	the equivalent of 2.25 full-time staff
Scope of the gain	receiving, picking, packing and shipping

The 15% is an assumption and, unusually, one this paper can say something about. Section 2.1 records that travel dominates picking time, so a labour gain of this size is plausible only if it comes substantially from travel — which means it depends on the slotting quality of Section 4.2 rather than on the software being installed. A facility that deploys the system and overrides directed putaway routinely should not expect it.

6 Discussion

6.1 What the ceiling actually is

The product's material locates the ceiling at three clients and two hundred thousand square feet. The floor area is a proxy; the binding variable is the number of distinct rule sets the facility must run at once.

One client is one set of putaway rules, one pick path, one packing flow. Three clients sharing docks and aisles is not three times that — it is three rule sets plus the interactions between them, because a shared resource must be allocated under rules that differ by the client claiming it. A system where client is a filter rather than a first-class dimension expresses those interactions as exceptions, and exceptions accumulate until somebody opens a spreadsheet.

This reframes the buying question. Not whether the system supports multiple clients, but whether client is a dimension of the allocation decision or a column on the report — and the way to find out is to ask for two clients with contradictory putaway rules competing for the same aisle.

A facility does not outgrow a warehouse system's capacity. It outgrows the number of simultaneous rule sets the system can express without exceptions.

6.2 Two count regimes, and what each can control

This library contains a paper on the annual physical count and this one on continuous counting, and they are not competitors. They control different things and a facility usually needs both.

Table 7. What each counting regime controls. The distinction is staleness against completeness, and neither substitutes for the other.

	Continuous cycle count	Periodic full count
Controls	expected staleness of a wrong record	completeness of coverage at a point in time
Coverage	biased toward what is picked	every location, including those nobody visits
Cost	absorbed into normal operations	operational shutdown
Blind spot	slow-moving and dormant stock	everything that happens between counts
Evidence value	operational — accuracy between counts	statutory — a position at a stated date

The blind spots are complementary, which is the point. Continuous counting is driven by picking activity, so dormant stock is exactly what it does not reach; a periodic count reaches everything and says nothing about the eleven months either side of it.

The practical consequence is that high record accuracy from continuous counting is not, on its own, an argument for abandoning the periodic count — it is an argument for being able to sample it. That transition is discussed in the companion paper on annual tagging and is gated on the accuracy figure being measured rather than asserted.

6.3 A capability reference framework for warehouse management systems

Table 8. Capability reference framework. Each dimension separates a warehouse management system from an inventory database with screens, and each is answerable by demonstration on a live floor rather than by reading a specification.

Dimension	Question the system must answer by demonstration
D1 Governed destinations	Attempt a putaway to a location that is not in the master. Is it accepted?
D2 Capacity as a guard	Put away into a location already at capacity. Refused, or warned and recorded?
D3 Type as a guard	Store a unit in a location typed for a different stock state. What happens?
D4 Suggested against used	Does the record keep both the location the system directed and the one chosen?
D5 Override cost visible	Can the system report the travel cost of last month's ad-hoc putaways?
D6 Count as observation	Does a count write an observation and a separate attributable adjustment, or overwrite the quantity?
D7 Staleness reported	How long, on average, does a wrong record survive before it is found?
D8 Contract stability	Did the same integration endpoint serve the last three clients unchanged?

D4 is the cheapest and most revealing. A system that discards its own suggestion cannot be evaluated, improved, or held to account for the slotting it produced.

6.4 Generalisability

The two-coordinate invariant generalises to any custody system where objects have a state and a place: tool cribs, evidence rooms, laboratory sample stores, container yards. The staleness characterisation of Section 4.3 generalises to every inspection regime where errors are discovered either on a schedule or on use, and the μ term is always concentrated on what is used.

What does not generalise is the onboarding figure. Six weeks to eight days is a comparison against one specific replaced system, and the size of the improvement is a property of what was there before rather than of what replaced it.

7 Threats to Validity and Limitations

1. The onboarding result is a single operator's experience. No number of clients onboarded, no complexity distribution and no definition of what onboarding includes is published, and the six-week baseline describes the system replaced rather than the category.
2. The captures do not show multi-client operation. The build observed is a manufacturing warehouse with three stock states and production planning; nothing in these figures demonstrates client isolation, shared-dock allocation or per-client rule sets.
3. The 99% accuracy claim on the product page has no baseline. The attributed 99.4% belongs to a deployment and to a different module; the two figures should not be read as one claim.
4. The modelled 15% labour gain is an assumption. Section 5.4 argues it depends on slotting quality rather than on installation, which makes it a claim about how the system is operated.
5. Slotting quality is not observable from a screenshot. The captures show that locations carry type and capacity; they do not show what the directed-putaway engine optimises or whether the objective of Equation (putaway) is the one implemented.
6. The staleness model assumes independent errors. Where a single mis-scan corrupts a batch of adjacent locations, the discovery process is correlated and expected staleness is longer than Equation (staleness) suggests.
7. Scan reliability is an unmeasured dependency. Every guard in Section 4.1 assumes the identifier scans; a degraded label converts a validated transition into a typed one with no change to the interface.
8. Cycle counting reaches what is picked. Dormant stock accumulates error at the rate it always did, and the accuracy figure a continuous regime reports is weighted toward the locations it visits.

The second limitation is the one to keep in view when reading this paper. The strongest published claim concerns multi-client operation and the strongest visual evidence concerns a manufacturing warehouse; they are different deployments, and this paper reports each only where its own source does.

8 Future Work

- Publishing override cost. Both the suggested and the used location are recorded, so the travel penalty of Equation (override) is computable today; reporting it would let a facility set a surge policy on evidence rather than on instinct.
- Staleness as a published metric. Reporting the mean interval between a record becoming wrong and being found would replace an accuracy percentage whose denominator is rarely stated.
- Velocity-thresholded override. Permitting ad-hoc putaway only below a pick-frequency threshold would keep the receiving-surge benefit while removing the expensive tail of it.
- An onboarding study with a distribution. Eight days from one operator is an anecdote; ten clients with their integration scope stated would be evidence, and the mechanism in Equation (onboard) predicts

what the shape should look like.

- Dormant-stock coverage. A count trigger driven by time since last observation, rather than by picking activity, would close the blind spot Section 6.2 identifies without reintroducing a freeze.

9 Conclusion

A warehouse management system is a ledger asserting one state and one location for every object, and every workflow it offers is a guarded transition on that pair. Stating it that narrowly makes the system testable: the four obligations of Section 4.1 are things a buyer can attempt to violate on a live floor in an afternoon.

Three results follow. A location record carrying a type and a capacity turns putaway into a feasibility problem rather than a suggestion, which is what allows the guard to refuse rather than warn. Permitting an ad-hoc override is a defensible trade whose price is computable from data the system already stores — and it is far more expensive on fast movers than on slow ones. And continuous counting controls expected staleness rather than count frequency, with its discovery rate concentrated exactly where errors are costly, which is the property no periodic count can reproduce.

The published onboarding result belongs to this module and its mechanism is arithmetic: mappings against a contract that does not move, rather than a build per client. The same deployment's accuracy and dock results belong to other modules, and this paper has left them there.

The framework of Section 6.3 is offered as the durable contribution. Its fourth dimension costs nothing to check and settles more than the rest: ask whether the system still knows where it told the operator to put the pallet.

Appendix A Nomenclature

Table 9. Symbols and abbreviations used in this paper.

Symbol / term	Meaning
U, S, L	The sets of tracked units, stock states and locations
$\sigma(u)$, $\lambda(u)$	The state and the location assigned to unit u
$v(u)$	Volume of unit u , in the capacity units of a location
$\kappa(l)$	Capacity carried on location l 's master record
$\tau(u, l)$	1 where location l 's type accepts unit u 's state, 0 otherwise
$f(u)$	Expected pick frequency of unit u
$d(l)$	Travel distance from the depot to location l
$x(u, l)$	Assignment variable: 1 where unit u is stored at location l
O	The set of putaways overridden by an operator in a period
Delta C	Total travel cost of those overrides
c	Count coverage rate — counts per location per period
μ	Rate at which picking activity raises a discrepancy count task
Theta	Record staleness — time from a record becoming wrong to it being found
a, b(k), m(k)	Fixed integration cost, bespoke build for client k , mapping for client k
RM / SFG / FG	Raw material, semi-finished goods, finished goods
GRN	Goods receipt note — the document a receiving transition resolves against

Appendix B Worked Numerical Examples

Appendix B.1 What one week of ad-hoc putaway costs

A receiving surge produces 120 overridden putaways in a week. 18 of them are fast movers with an expected pick frequency of 40 picks per month; the remaining 102 are slow movers at 1.5 picks per month. In each case the chosen location is on average 22 metres further from the depot than the directed one.

Applying Equation (override) over a month: the fast movers contribute $18 \times 40 \times 22 = 15,840$ metres of additional travel; the slow movers contribute $102 \times 1.5 \times 22 = 3,366$. The total is 19,206 metres a month, and the mean cost per override is 160 metres.

The distribution is the finding. Fifteen per cent of the overrides generate eighty-two per cent of the cost. A policy permitting override only above a slow-moving threshold would keep 102 of the 120 overrides — almost all the receiving-surge relief — while removing four fifths of the penalty.

Both distances are in the record, because the system stores what it suggested. Without that field this calculation is impossible and the surge policy is set by argument.

Appendix B.2 Two counting regimes on the same warehouse

A facility has 4,000 locations. Under an annual freeze count, coverage is one per location per year.

Applying Equation (staleness): $c = 1$ per location-year, so expected staleness is 0.5 years — about six months. An error introduced in March is found, on average, in September, by which time the shift, the operator and the movement that caused it are unrecoverable.

Now run continuous counting at 60 count tasks per working day over 250 days: $c = 15,000 / 4,000 = 3.75$ per location-year, giving 0.133 years — about seven weeks — from scheduled counting alone. Add picker-triggered discovery on the fast-moving third of locations at $\mu = 20$ per location-year, and staleness on those falls to $1 / (2 \times 3.75 + 20) = 0.036$ years, or under two weeks.

The asymmetry is the design working as intended. Errors in locations that are picked from constantly are found in days; errors in locations nobody touches persist — and cost nothing until somebody does touch them. What the regime cannot do is find the dormant error before the periodic count, which is why Section 6.2 keeps both.

Appendix B.3 Where the onboarding saving actually comes from

An operator onboards four clients a year. Under per-client pipelines each takes 30 working days, of which roughly 6 are scoping common to any approach and 24 are the bespoke build.

Applying Equation (onboard): pipeline effort is $4 \times 30 = 120$ days a year. Against a stable contract with a one-off interface cost of 40 days already paid, a mapping takes 6 days of scoping plus 2 of field mapping, so $4 \times 8 = 32$ days a year.

Two readings follow. The steady-state saving is 88 days a year, and the interface pays for itself inside the first six clients. And the published eight-day figure is consistent with this shape — it is the mapping term, not the whole engagement, which is worth knowing before quoting it to a client.

Data provenance

Every quantitative claim in this paper resolves to one of the entries below. Figures marked as modelled estimates are not deployment measurements; their assumptions are stated.

Metric	Reported value	Provenance
New client onboarding time	8 days (from 6 weeks)	Operator-reported — Tier 1 Manufacturers (3PL, Madhya Pradesh, India) /case-studies/warehouse — REST APIs replaced per-client custom data pipelines.
Location record granularity in the location master	zone-rack-bin code, typed, with capacity and current stock	Product specification /products/wms — Read from the location master captured on the product page; each location can also generate its own barcode.
Stock states tracked as separate positions on the management dashboard	raw material, semi-finished, finished goods	Product specification /products/wms — Read from the management dashboard captured on the product page; the build shown is a manufacturing warehouse spanning production, not a pure distribution deployment.
Inventory accuracy the product page claims	99% or better	Product specification /products/wms — A product-page benefit claim without a baseline or sample; the case-study figure of 99.4% is a separate, attributed measurement.
Labour hours recovered across receiving, picking, packing and shipping	15% of 15 staff (modelled)	Modelled estimate 15 warehouse staff in scope; 15% labour gain attributed to the module; Model and defaults published in src/data/roiModels.ts
Inventory accuracy	99.4% (from 91%)	Operator-reported — Tier 1 Manufacturers (3PL, Madhya Pradesh, India) /case-studies/warehouse — Continuous exception-triggered cycle counting replaced scheduled freeze counts.
Dock idle time	38% reduction	Operator-reported — Tier 1 Manufacturers (3PL, Madhya Pradesh, India) /case-studies/warehouse

References

- [1] de Koster, R., Le-Duc, T., and Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2), 481-501.
- [2] Roodbergen, K. J., and de Koster, R. (2001). Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39(9), 1865-1883.
- [3] Petersen, C. G. (1997). An evaluation of order picking routing policies. *International Journal of Operations & Production Management*, 17(11), 1098-1111.
- [4] Gu, J., Goetschalckx, M., and McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177(1), 1-21.

- [5] H. F. Dickie (1951). ABC Inventory Analysis Shoots for Dollars, Not Pennies. *Factory Management and Maintenance*, vol. 109, no. 7, July 1951, pp. 92-94; the origin of ABC classification, applying Pareto and Lorenz to stock value.
- [6] Little, J. D. C. (1961). A proof for the queuing formula: $L = \lambda W$. *Operations Research*, 9(3), 383-387.
- [7] F. W. Harris (1913). How Many Parts to Make at Once. *Factory, The Magazine of Management*, vol. 10, no. 2, pp. 135-136, 152; the first statement of the economic order quantity, reprinted in *Operations Research* 38(6), 1990, pp. 947-950. <https://pubsonline.informs.org/doi/10.1287/opre.38.6.947>
- [8] International Organization for Standardization / International Electrotechnical Commission (2024). ISO/IEC 19987: Information technology - EPC Information Services (EPCIS) Standard, version 2.0. ISO/IEC, Geneva. <https://www.iso.org/standard/85557.html>
- [9] International Organization for Standardization / International Electrotechnical Commission (2024). ISO/IEC 19988:2024 - Information technology - GS1 Core Business Vocabulary (CBV). ISO/IEC, Geneva, edition 3, March 2024; publishes GS1 CBV 2.0, ratified June 2022, which fixes the vocabularies for the EPCIS business step, disposition and business transaction type fields. <https://www.iso.org/standard/85558.html>
- [10] GS1 (2024). GS1 General Specifications. GS1 AISBL; defines the GTIN, SSCC and GLN identification keys. <https://ref.gs1.org/standards/genspecs/>
- [11] GS1 (2024). Serial Shipping Container Code (SSCC). GS1 identification key series; an 18-digit key for a logistic unit, carried under application identifier 00. <https://www.gs1.org/standards/id-keys/sscc>
- [12] GS1 (2023). GS1 Logistic Label Guideline. GS1 AISBL; the SSCC is the only mandatory field on a GS1 logistic label. <https://www.gs1.org/standards/gs1-logistic-label-guideline/1-3>
- [13] International Organization for Standardization / International Electrotechnical Commission (2014). ISO/IEC 15459-1: Information technology - Automatic identification and data capture techniques - Unique identification - Part 1: Individual transport units. ISO/IEC, Geneva. <https://www.iso.org/standard/54779.html>
- [14] International Organization for Standardization (2014). ISO 22400-1: Automation systems and integration - Key performance indicators (KPIs) for manufacturing operations management - Part 1: Overview, concepts and terminology. ISO, Geneva. <https://www.iso.org/standard/56847.html>
- [15] International Organization for Standardization (2014). ISO 22400-2: Automation systems and integration - Key performance indicators (KPIs) for manufacturing operations management - Part 2: Definitions and descriptions. ISO, Geneva; defines OEE as availability x effectiveness x quality ratio. <https://www.iso.org/standard/54497.html>
- [16] International Electrotechnical Commission / International Society of Automation (2025). Enterprise-control system integration - Part 1: Models and terminology. IEC 62264-1; ANSI/ISA-95.00.01-2025 (IEC 62264-1 Mod).
- [17] International Society of Automation (2013). Enterprise-control system integration - Part 3: Activity models of manufacturing operations management. ANSI/ISA-95.00.03-2013 (IEC 62264-3 Modified).
- [18] International Organization for Standardization (2022). ISO 28000: Security and resilience - Security management systems - Requirements. ISO, Geneva; published 15 March 2022. <https://www.iso.org/standard/79612.html>
- [19] International Organization for Standardization (2023). ISO 5725-1: Accuracy (trueness and precision) of measurement methods and results - Part 1: General principles and definitions. ISO, Geneva (superseding ISO 5725-1:1994).

- [20] Kagermann, H., Wahlster, W., and Helbig, J. (2013). Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group, acatech - National Academy of Science and Engineering. https://www.acatech.de/wp-content/uploads/2018/03/Final_report__Industrie_4.0_accessible.pdf