

Prozess-Landkarte Lager & Produktion

Stand 2026-09-03 · Programm Replikation · Meilenstein R4 · Epic „Scanner-App neu“ · Task E1 6a99329e6edab122065e8207 · alle Angaben gegen den Arbeitsbaum und die Produktions-Datenbank gemessen.

Diese Karte steht vor jedem Scanner-Bau (CEO-Entscheid 2026-09-03). Sie beantwortet nicht „welcher Screen fehlt“, sondern wo bricht die Kette — denn die Screens sind grösstenteils da.

Das Ergebnis in fünf Sätzen

Die Flächen sind gebaut. Der Scanner hat neun buchende Prozess-Screens, das Terminal siebzehn Seiten. Der Epic-Task „Prozess-Screens nachziehen“ ist damit ein Schliess-, kein Bau-Auftrag.

Die Buchung ist die Engstelle. Von allen Lager-Ketten sind genau drei bestandswirksam: Stocks-Sammelaktionen, MaterialCommissionings.consume, InventoryAdjustments.apply. Einlagern bucht nicht ein. Retoure bucht nicht zurück. Packen bucht nicht aus.

Keine der 13 WMS-Entitäten ist deklariert. Alle 13 haben hand-gebaute Aktions-Router und null entityActions im Register — die Aktionen existieren, die Plattform kennt sie nicht.

Die Kette endet nach dem Pick. Verpackung und Versand sind in beiden Client-Diensten fertig verdrahtet und haben in keinem Client eine Fläche.

Niemand wird benachrichtigt. Von 151 Benachrichtigungs-Regeln trägt keine einzige eine Lager-Entität.

Legende der Spalten

| Spalte | Bedeutung | |---|---| | Schritt | fachlicher Vorgang | | Rolle | Werker · Lagerleiter · Einkäufer · Produktionsleiter · System | | Gerät/Fläche | Scanner-Route · Terminal-Seite · PWA-Zustand · Wallboard-Widget · keine | | Buchung | Entität + Aktion + Recht; fett = bestandswirksam (Stocks/ StockLogs) | | Benachrichtigung | notificationpolicies-Eintrag oder keine (gemessen, nicht geraten) | | Ist-Fläche | Datei:Zeile — der Beleg, dass es die Fläche wirklich gibt | | Lücke | was fehlt, mit Träger-Zuordnung |

Ø=Ÿ4 Rechte-Hinweis für alle WMS-Ketten: jede Aktion läuft auf dem Basis-Modulrecht (warehouse.<entität>), nicht auf einem abgeleiteten ...actions.<name> — weil die Aktion nicht deklariert ist. Eine feinere Rechtevergabe ist heute strukturell unmöglich.

& Alias-Hinweis: /stocks/book/incoming, /stocks/book/transfer und /stocks/count sind bewusste Aliase auf die actions/-Form (routerStocks.ts:1000-1003) — Scanner und Terminal nutzen verschiedene Schreibweisen derselben Route, das ist kein Defekt.

Kette 1 — Wareneingang (Lieferung !' Bestand)

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---|---|

Lieferung avisieren | Einkäufer | PWA purchase.supplierorders | SupplierOrders !' order · purchase.supplierorders | approval_requested/approved/rejected | modelsAndRights/purchase.ts:169 | — | | Anlieferung am Tor | Lagerleiter | keine | DockAppointments !' arrive · warehouse.dockappointments | keine | routerDockAppointments.ts:243 | Fläche fehlt ganz !' warehouse.think-tank-storage-structure (R4) | | Lieferschein scannen | Werker | Scanner receive | SupplierOrders lesen | keine | goods-receive.component.ts:451,467 | — | | Ware erfassen | Werker | Scanner receive · Terminal warehouse/receive | Stocks !' book-incoming (StockLogs Zugang + Stocks \$inc upsert) | keine | goods-receive.component.ts:577 · Terminal receive.component.ts:441 | Benachrichtigung !' automation.order-chain-notifications (R6) | | Wareneingang belegen | System | Terminal warehouse/goods-receipts (nur Liste) | GoodsReceipts — kein Router, keine Aktion | keine | Terminal goods-receipts.component.ts:97 | Ø=Ý4 Scanner erzeugt keinen GoodsReceipt !' M-B-Task 6a941ea4fad888ed370600a9 | | Menge zurückmelden | System | — | SupplierOrder-Zeile qtyReceived (Roll-up in book-incoming) | keine | routerStocks.ts:365 | — |

Positiv-Kontrolle der Kette: book-incoming schreibt nachweislich beides (StockLogs :365, Stocks \$inc upsert) — die Null-Aussagen der Nachbar-Schritte sind damit belastbar.

Ø=Ý4 Der Bruch: der Scanner bucht den Bestand, legt aber keinen Wareneingangs-Beleg an. Die Entität GoodsReceipts hat weder entityActions (purchase.ts:9) noch einen Router in warehouse-api; sie wird nur gelesen. Wer den Wareneingang später prüfen will, findet die Buchung im StockLog, aber keinen Beleg mit Position und Foto.

Kette 2 — Einlagern (Auftrag !' Lagerplatz)

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---| | Einlager-Auftrag entsteht | System | PWA warehouse.putawayorders (Liste) | PutAwayOrders anlegen · warehouse.putawayorders | keine | ayounestates warehouse.putawayorders | kein Erzeuger gemessen !' warehouse.pick-to-workcenter (R4) | | Auftrag holen | Werker | Scanner putaway | PutAwayOrders lesen | keine | putaway.component.ts:218 | — | | Lagerort scannen | Werker | Scanner putaway | Auflöser POST scan.ayoune.app/ | — | putaway.component.ts:296 | — | | Menge bestätigen | Werker | Scanner putaway | PutAwayOrders !' complete · warehouse.putawayorders | keine | putaway.component.ts:319 · routerPutAwayOrders.ts:145 | Ø=Ý4 bucht den Bestand NICHT | | Bestand am Platz | System | — | — | — | — | Ø=Ý4 fehlt vollständig |

Ø=Ý4 Der schwerste Bruch der ganzen Karte. PutAwayOrders.complete fasst Stocks und StockLogs null-mal an (gemessen; Positiv-Kontrolle: routerStocks.ts trägt 18 Treffer derselben Suche). Der Werker quittiert das Einlagern, der Bestand wandert nicht auf den Lagerplatz. Bestand entsteht heute ausschliesslich über book-incoming — also am Wareneingang, nicht am Platz.

Träger: warehouse.think-tank-storage-structure (R4, plan:99 — „Bestände + Mindestbestände“).

Kette 3 — Umlagern (Platz !' Platz)

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---| | Quelle scannen | Werker | Scanner transfer · Terminal warehouse/transfer | Stocks lesen | — | transfer.component.ts:315 | — | | Ziel scannen | Werker | Scanner transfer · Terminal warehouse/transfer | StorageAreas lesen | — | transfer.component.ts:357 | — | | Umlagerung buchen | Werker |

Scanner transfer · Terminal warehouse/transfer | Stocks !' transfer (2 Bestandszeilen + 2 StockLogs Umlagerung, \$gte-Guard, Kompensation) | keine | transfer.component.ts:419 · routerStocks.ts:654 | — | | Transportauftrag | Lagerleiter | keine | TransportOrders !' confirm/ready/dispatch/deliver · warehouse.transportorders | keine | routerTransportOrders.ts:211-214 | Fläche fehlt ganz; 4 Aktionen ohne Aufrufer |

Ø=ßâ Die einzige Kette, die technisch sauber durchläuft. transfer bucht beide Seiten, hält einen \$gte-Guard gegen Unterdeckung im Filter und kompensiert bei Fehler — die reifste Buchung im Bestand.

& TransportOrders ist der Gegensatz: vier Aktions-Routen, keine Bestandswirkung, keine Fläche in keinem Client (im Scanner-Dienst existiert nicht einmal eine Methode dafür).

Kette 4 — Inventur und Korrektur

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---|---| | Zähl-Runde starten | Lagerleiter | Scanner count | StorageAreas + Stocks lesen | — | cycle-count.component.ts:246,256 | — | | Zählen | Werker | Scanner count | Stocks !' count (runCycleCount: StockLogs + Abgleich auf Zählwert) | keine | cycle-count.component.ts:289 · routerStocks.ts:1002 | — | | Differenz erfassen | Werker | Scanner count !' adjust | InventoryAdjustments anlegen · warehouse.inventoryadjustments | keine | cycle-count.component.ts:266 · adjust.component.ts:412 | — | | Zur Freigabe geben | Werker | Scanner adjust | InventoryAdjustments !' submit | keine | adjust.component.ts:470 · routerInventoryAdjustments.ts:330 | Ø=Ÿ4 der Freigeber erfährt nichts | | Freigeben/Ablehnen | Lagerleiter | Scanner adjust | InventoryAdjustments !' approve/reject | keine | adjust.component.ts:494,500 | Benachrichtigung !' automation.order-chain-notifications (R6) | | Korrektur buchen | System | Scanner adjust | InventoryAdjustments !' apply (runAdjustmentApply !' reconcileStockBucket: StockLogs + Stocks) | keine | adjust.component.ts:548 · routerInventoryAdjustments.ts:333 | — |

Ø=ßâ Die einzige Kette mit einer vollständigen Freigabe-Strecke (submit !' approve/reject !' apply) und einer echten Bestandsbuchung am Ende.

Ø=Ÿ4 Und genau sie zeigt die Benachrichtigungs-Lücke am schärfsten: eine Bestandskorrektur wartet auf eine Freigabe, und niemand wird informiert. Der Freigeber müsste die Liste von sich aus öffnen. Es gibt dafür weder eine Regel (0 von 151) noch ein Wallboard-Widget (0 von 21).

Kette 5 — Pick (Bedarf !' Entnahme)

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---|---| | Welle bilden | Lagerleiter | Scanner pick | PickWaves !' release · warehouse.pickwaves | keine | routerPickWaves.ts:212 | release hat keinen Aufrufer | | Welle starten | Werker | Scanner pick | PickWaves !' start | keine | pick-list.component.ts:644 | — | | Pickliste übernehmen | Werker | Scanner pick | PickLists !' assign · warehouse.picklists | keine | pick-list.component.ts:389 | Ø=Ÿ4 der Werker erfährt nicht, dass eine Liste bereitliegt | | Position picken | Werker | Scanner pick | PickLists !' confirm-position (nur Positionsfelder + Roll-up) | keine | routerPickLists.ts:319 | Ø=Ÿ4 bucht den Bestand NICHT | | Reservierung erfüllen | System | Scanner pick | StockReservations !' fulfill (Stocks nur lesend) | keine | pick-list.component.ts:653 · routerStockReservations.ts:378 | Ø=Ÿ4 bucht den Bestand NICHT | | Liste abschliessen | Werker | Scanner pick | PickLists !' complete | keine | pick-list.component.ts:416 | Ø=Ÿ4 bucht

den Bestand NICHT | | Bestand entnehmen | System | — | Stocks !' book-outgoing existiert, wird von der Kette nicht gerufen | keine | routerStocks.ts:886 | Ø=Ý4 Bruch |

Ø=Ý4 Der zweite schwere Bruch. Die gesamte Pick-Kette ist bestandsneutral: confirm-position, complete und fulfill fassen Stocks/StockLogs null-mal an. Der Ausgangs-Weg POST /stocks/actions/book-outgoing existiert (routerStocks.ts:886, schreibt den Bestand ohne upsert plus einen StockLog vom Typ Entnahme) — er wird von keiner Pick-Route und keinem Client gerufen.

& Nebenbefund zur Reihenfolge: book-outgoing schreibt erst den Bestand, dann das Journal (:886), während book-incoming und reconcileStockBucket es umgekehrt tun (Journal zuerst, stockReconcile.ts:84 vor :107). Bei einem Abbruch dazwischen sind die zwei Wege verschieden fehlerhaft. Das gehört benannt, bevor book-outgoing scharfgeschaltet wird.

Träger: warehouse.pick-commission-ship (R6, plan:134).

Kette 6 — Kommission an den Arbeitsplatz

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---| |
Bedarf je Schritt | System | — | MaterialCommissionings anlegen · warehouse.materialcommissionings
| keine | warehouse.ts:53 | Erzeuger nicht gemessen !' purchase.bom-explosion-demand (R6) | | Material
bereitstellen | Werker | Scanner commission · Terminal production/step | MaterialCommissionings !'
deliver | keine | commission.component.ts:287 · Terminal step.component.ts:698 | bucht nicht (als reine
Bereitstellung richtig) | | An Arbeitsplatz binden | Werker | Terminal production/step |
MaterialCommissionings lesen (Filter auf den Produktionsauftrag) | — | Terminal
step.component.ts:696 | — | | Verbrauch buchen | System | Terminal production/step (automatisch bei
100 % Fortschritt) | MaterialCommissionings !' consume (runMaterialConsume !' StockLogs + Stocks +
Status auf verbraucht) | keine | routerMaterialCommissionings.ts:272 · materialConsume.ts:212 | — |

Ø=ßâ Die einzige Kette, die zwischen zwei Domänen sauber durchläuft — und die einzige, in der eine Buchung automatisch aus einem Fachereignis folgt statt aus einem Knopfdruck.

& Bemerkenswert und richtig gebaut: production-api schreibt nie selbst auf Stocks/Lots/StockLogs (warehouseClient.ts:6-8), sondern ruft warehouse-api mit dem weitergereichten Bearer-Token des Aufrufers (:53-58), ausdrücklich nicht über den internen Dienst-Weg. Damit bucht der Werker unter seiner eigenen Kennung.

Kette 7 — Produktion und Materialverbrauch

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---| |
Auftrag starten | Produktionsleiter | PWA production.productionorders | ProductionOrders !' start ·
production.productionorders | ' ProductionOrders / started | production.ts:69 | — | | Arbeitsplatz belegen
| Werker | Terminal production/workcenter | WorkCenters !' set-occupied | keine | Terminal
workcenter.component.ts:211 | — | | Schritt beenden | Werker | Terminal production/step | hand-
gebaute MES-Route stop !' ProductionOrders | Ereignis stopped | routerProductionSteps.ts:432 | — | |
Zählerstand prüfen | System | Terminal production/step | — (bricht vor jedem Schreibvorgang ab) | — |
applyStepTransition.ts:413 | — | | Fortschritt hochrollen | System | — | ProductionOrders Roll-up
Schritt!'Position!'Auftrag | keine | rollupOrderProgress.ts | — | | Material verbrauchen | System | — (nur bei
100 % Fortschritt) | MaterialCommissionings !' consume | keine | routerProductionSteps.ts:358-375 | — |

| Zeit buchen | Werker | Terminal production/step | POST production-api/worklogs/actions/book | keine | Terminal step.component.ts:1443 | — | | Auftrag abschliessen | Produktionsleiter | PWA production.productionorders | ProductionOrders !' complete | ' ProductionOrders / completed | production.ts:69 | — |

Ø=ß Die reifste Kette der Plattform — und die einzige mit Benachrichtigungen (zwei Regeln, started und completed, in drei Mandanten-Kopien).

& Zwei Fallstricke, die in die Karte gehören:

Der Materialverbrauch hängt an Geschäftsnummern, nicht an Array-Positionen (warehouseConsume.ts:10-20). Fehlen sie, wird der Verbrauch still übersprungen, ohne Fehler.

Ein Buchungs-Problem wird zur Warnung, nie zum Fehler — foldConsumeOutcome (warehouseConsume.ts:69-72) übersetzt selbst eine Unterdeckung in eine Warnung, damit ein produzierter Schritt nie als Fehlschlag erscheint. Das ist fachlich richtig und heisst zugleich: eine fehlgeschlagene Materialbuchung ist am Terminal nicht als Fehler sichtbar.

Kette 8 — Verpackung und Versand

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---| | Packauftrag entsteht | System | keine | PackConfirmations anlegen · warehouse.packconfirmations | keine | routerPackConfirmations.ts | Ø=Ý4 kein Erzeuger, keine Fläche | | Karton befüllen | Werker | keine | — | keine | — | Ø=Ý4 fehlt vollständig | | Karton versiegeln | Werker | keine | PackConfirmations !' seal-box | keine | routerPackConfirmations.ts:215 | Ø=Ý4 Aktion ohne Aufrufer in beiden Clients | | Packen abschliessen | Werker | keine | PackConfirmations !' complete | keine | routerPackConfirmations.ts:216 | Ø=Ý4 Aktion ohne Aufrufer, bucht nicht aus | | Versandkarton verwalten | Lagerleiter | Terminal warehouse/shipping-boxes | ShippingBoxes (purchase-api) CRUD | keine | Terminal shipping-boxes.component.ts:231,248 | die einzige Fläche der ganzen Kette | | Versand | System | — | Shippings | ' outfordelivery · delivered · exception | notificationpolicies | die Benachrichtigungen sind da, der Vorgang davor nicht |

Ø=Ý4 Die Kette endet nach dem Pick. Alle drei Packbestätigungs-Methoden sind in beiden Client-Diensten deklariert (createPackConfirmation, sealBox, completePackConfirmation) und haben in keinem Client eine Fläche. Der einzige gebaute Teil ist die Verwaltung der Versandkartons — als Stammdaten, nicht als Vorgang.

& Der Kontrast ist die Aussage: für den Versand danach existieren drei Benachrichtigungs-Regeln (Shippings), für das Packen davor keine einzige — und keine Fläche. Der Übergang vom Lager zum Versand ist die grösste zusammenhängende Lücke der Karte.

Träger: warehouse.pick-commission-ship (R6, plan:134).

Kette 9 — Retoure

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---| | Retoure anmelden | System | keine | ReturnOrders anlegen · warehouse.returnorders | keine | warehouse.ts:119 | kein Erzeuger gemessen | | Prüfen und freigeben | Lagerleiter | Scanner returns | ReturnOrders !' approve/reject | keine | returns.component.ts:300,311 | — | | Ware annehmen | Werker |

Scanner returns | ReturnOrders !' receive | keine | returns.component.ts:334 · routerReturnOrders.ts:273
 | Ø=Ý4 bucht den Bestand NICHT zurück | | Zustand begutachten | Werker | Scanner returns | ReturnOrders !'
 inspect | keine | returns.component.ts:340 | — | | Ins Sperrlager | Werker | keine | — | keine | — | Ø=Ý4 fehlt
 vollständig | | Abschliessen | Werker | Scanner returns | ReturnOrders !' complete | keine |
 returns.component.ts:403 | Ø=Ý4 bucht nicht |

Ø=Ý4 Der dritte schwere Bruch. routerReturnOrders.ts fasst Stocks/StockLogs über alle sechs Aktionen null-mal an (Positiv-Kontrolle: routerStocks.ts 18 Treffer derselben Suche). Eine zurückgenommene Ware wird begutachtet, freigegeben und abgeschlossen — und ist im Bestand nie angekommen. Ein Sperrlager-Begriff existiert nirgends im Bestand.

Träger: neu — siehe Lücken-Register L9.

Kette 10 — Nachschub

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---|---| |
 Mindestbestand unterschritten | System | keine | — | keine | — | Ø=Ý4 kein Erzeuger !' purchase.reorder-
 demand (R4, plan:103) | | Auftrag übernehmen | Werker | Scanner replenish | ReplenishmentOrders !'
 assign | keine | replenish.component.ts:209 | — | | Nachfüllen | Werker | Scanner replenish |
 ReplenishmentOrders !' complete | keine | replenish.component.ts:251 ·
 routerReplenishmentOrders.ts:169 | Ø=Ý4 bucht den Bestand NICHT um |

Ø=Ý4 Ein Nachschub-Vorgang, der den Bestand nicht vom Reserve- auf den Kommissionierplatz bewegt, ist eine Quittung ohne Wirkung. Der Umlager-Weg dafür existiert (Stocks !' transfer, Kette 3) und wird nicht gerufen.

Kette 11 — Rückstand

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---|---| |
 Rückstand entsteht | System | PWA warehouse.backorders (Liste) | Backorders anlegen ·
 warehouse.backorders | keine | warehouse.ts:9 | kein Erzeuger gemessen | | Rückstand auflösen |
 Lagerleiter | keine | Backorders !' fulfill (Stocks nur lesend) | keine | routerBackorders.ts:308 | Ø=Ý4 Aktion
 ohne Aufrufer; Methode im Scanner-Dienst deklariert, von keiner Seite gerufen | | Rückstand stornieren
 | Lagerleiter | keine | Backorders !' cancel | keine | routerBackorders.ts:307 | Ø=Ý4 Aktion ohne Aufrufer |

Kette 12 — Dock und Anlieferung

| Schritt | Rolle | Gerät/Fläche | Buchung | Benachrichtigung | Ist-Fläche | Lücke | |---|---|---|---|---|---|---| |
 Zeitfenster buchen | Einkäufer | keine | DockAppointments anlegen | keine | warehouse.ts:20 | Ø=Ý4 keine
 Fläche | | Ankunft bestätigen | Lagerleiter | keine | DockAppointments !' confirm/arrive | keine |
 routerDockAppointments.ts:242,243 | Ø=Ý4 im Terminal-Dienst deklariert (confirmDockAppointment), ohne
 Aufrufer; im Scanner-Dienst gar nicht vorhanden | | Verladen | Werker | keine | DockAppointments !' start-
 loading/complete | keine | routerDockAppointments.ts:244,245 | Ø=Ý4 keine Fläche | | Tor belegen/freigeben
 | Lagerleiter | keine | Docks !' occupy/release | keine | routerDocks.ts:154,155 | Ø=Ý4 keine Fläche | |
 Nichterscheinen | Lagerleiter | keine | DockAppointments !' no-show | keine |
 routerDockAppointments.ts:246 | Ø=Ý4 keine Fläche |

Ø=Ý4 Die vollständig unerreichbare Kette. Elf Aktions-Routen über zwei Entitäten, keine einzige Fläche in Scanner, Terminal, PWA oder Wallboard. Für Think Tank Tegernsee heute ohne Bedarf — sie gehört in die Karte, weil sie sonst bei der nächsten Planung als „fehlt“ neu erfunden wird.

Lücken-Register — jede Lücke mit genau einem Träger

Träger-Arten: F = R-Programm-Feature · E = Task im Epic „Scanner-App neu“ · M = Task in den M-B-Sprints „Lager 1/2“ · N = neu (nur mit Begründung, CEO-Freigabe offen).

| # | Lücke | Kette | Träger | Art | |---|---|---|---|---| | L1 | Scanner bucht den Bestand, legt aber keinen Wareneingangs-Beleg an | 1 | Task 6a941ea4fad888ed370600a9 („Wareneingang im Scanner erzeugt einen GoodsReceipt mit Positionen und Foto“) | M | | L2 | PutAwayOrders.complete bucht den Bestand nicht auf den Platz | 2 | warehouse.think-tank-storage-structure (R4, plan:99) | F | | L3 | Pick-Kette vollständig bestandsneutral; book-outgoing existiert und wird nie gerufen — inkl. der abweichenden Schreib-Reihenfolge dort | 5 | warehouse.pick-commission-ship (R6, plan:134) | F | | L4 | Verpackung/Versand ohne jede Fläche, drei Aktionen ohne Aufrufer | 8 | warehouse.pick-commission-ship (R6, plan:134) | F | | L5 | ReturnOrders.receive bucht nicht zurück; kein Sperrlager-Begriff im Bestand | 9 | neu — die R6-Kette endet beim Versand, die Retoure ist der Rückweg und in keinem Feature enthalten (geprüft gegen alle 23) | N | | L6 | ReplenishmentOrders.complete bucht den Bestand nicht um | 10 | purchase.reorder-demand (R4, plan:103 — Mindestbestand! Bedarf) | F | | L7 | Null Benachrichtigungen für die gesamte WMS-Achse (0 von 151 Regeln) | 1,4,5,8,9,10 | automation.order-chain-notifications (R6, plan:136) | F | | L8 | Alle 13 WMS-Entitäten ohne entityActions — Prozess ohne Deklaration | alle | neu — Fundament-Arbeit, in keinem Feature und keinem Task enthalten (geprüft gegen 23 F, 8 E, 15 M) | N | | L9 | Kein Einstieg für Lager und Lagerorte (keine Liste, kein Baum) | alle | Task 6a9932b26edab122065e84f4 („Einstieg Lager & Lagerorte verwalten“) | E | | L10 | Raster-Generator-Oberfläche existiert, ist aber ohne Route unerreichbar | 2 | warehouse.scanner-grid-ui (R4, plan:96) | F | | L11 | storage-area-detail liest keinen Scan — die einzige Lagerort-Fläche | 2 | Task 6a9932a16edab122065e8293 (Scan-Bus auf alle Seiten) | E | | L12 | Einstellungen sind sieben Listenpunkte ohne Verhalten | alle | Task 6a9932b76edab122065e8580 | E | | L13 | Dock/Anlieferung: 11 Aktionen über 2 Entitäten, null Flächen | 12 | zurückgestellt — für Think Tank Tegernsee heute ohne Bedarf; in der Karte, damit sie nicht neu erfunden wird | — | | L14 | Wallboard: 20 von 21 Endpunkten produktionsbezogen, für Lager genau einer (als „legacy“ markiert), kein Widget für Bestände/Picklisten | alle | reporting.production-floor-wallboards (R7, plan:155) | F | | L15 | TransportOrders/Backorders: Aktionen ohne jeden Aufrufer | 3,11 | purchase.reorder-demand (R4) für Backorders; TransportOrders mit L13 zurückgestellt | F |

Bilanz: 13 Lücken zugeordnet · 2 neu · 1 zurückgestellt. Die zwei neuen (L5, L8) sind gegen alle 23 Features, 8 Epic-Tasks und 15 M-B-Tasks geprüft. Ø=Ý4 Kein PM-Task angelegt — das braucht die CEO-Freigabe (Brief-Constraint).

& Beobachtung ohne Träger (gehört in den SDUI-Strang, nicht hierher): das Modul warehouse trägt 13 reine Listen-Zustände und keinen einzigen .edit.*-Detail-Reiter, während purchase 13 und production 22 tragen. Wer in der PWA einen Wareneingang oder eine Pickliste öffnen will, findet keine Bearbeitungsfläche.

Bau-Reihenfolge für Task 6a9932ba6edab122065e860c (REP3)

Ø=Ý4 Der Auftrag jenes Tasks lautet „Prozess-Screens nachziehen“ — gemessen ist er ein SCHLIESS-, kein Bau-Auftrag. Alle sieben genannten Screens (Wareneingang, Einlagern, Umlagern, Inventur, Pick, Kommission, Retoure) existieren im Scanner und rufen echte Aktions-Routen. Was fehlt, ist die Buchung dahinter.

Ø=ßâ Und die Buchungswege existieren ebenfalls alle — book-incoming, book-outgoing, transfer, count, apply, consume sind gebaut und getestet. Es ist Verdrahtung, kein Neubau.

Kriterium je Position: (a) Buchungsweg im Backend vorhanden · (b) Scan-Grundfunktion trägt den Schritt · (c) kein R4-Feature liefert den Screen ohnehin.

1. Die 13 Entitäten deklarieren (L8) — Fundament, vor allem anderen

Ohne entityActions im Register gibt es keine SDUI-Schaltfläche, keinen Eintrag in der Scan-Aktionsliste und kein abgeleitetes Recht — heute läuft jede der ~50 Aktionen auf dem Basis-Modulrecht. Jede folgende Position wird billiger, wenn das zuerst steht. & Die Stocks-Buchungen liegen auf Sammlungs-Ebene (/stocks/actions/*) und brauchen deshalb availability:'collection' — der Routen-Erzeuger kennt diese Form (belegt in CLAUDE.md, Kaskade 61). (a) ' · (b) entfällt · (c) '

2. Einlagern bucht ein (L2)

Ohne Bestand am Lagerplatz ist jede Folgekette wertlos — Pick, Nachschub und Inventur rechnen alle gegen einen Bestand, der heute nur am Wareneingang entsteht. Weg: PutAwayOrders.complete ruft Stocks !' transfer (Eingangszone !' Zielplatz). (a) ' · (b) ' (putaway scannt Platz und Menge) · (c) '

3. Einstieg Lager und Lagerorte (L9, L10, L11)

Der CEO-Befund Nr. 1 und zugleich die Voraussetzung für Position 2: man muss einen Lagerplatz sehen, auswählen und anschauen können. Die Raster-Oberfläche existiert bereits im Zusatzblatt und braucht nur einen Routen-Einstieg. (a) ' (generate-grid, print-cell-labels) · (b) & heute nein — storage-area-detail liest keinen Scan, das schliesst L11 mit · (c) & Überschneidung mit warehouse.scanner-grid-ui (R4) — dort bauen, hier nur verlinken, sonst doppelt.

4. Pick bucht aus (L3)

book-outgoing scharfschalten aus PickLists.complete. & Vorher die Schreib-Reihenfolge angleichen (heute Bestand vor Journal, überall sonst Journal vor Bestand) — sonst wird eine bestehende Abweichung zum ersten Mal wirksam. (a) ' · (b) ' · (c) '

5. Packen und Versand (L4)

Die erste Position, die eine neue Fläche braucht (Packauftrag, Karton befüllen, versiegeln). Die drei Aktionen sind gebaut, book-outgoing liefert die Ausbuchung, und die Benachrichtigungen für den Versand danach existieren bereits. (a) ' · (b) ' (Karton- und Artikel-Scan) · (c) '

6. Retoure bucht zurück (L5)

Rückweg über book-incoming auf einen Sperrbereich. & Der Sperrlager-Begriff existiert nirgends — das ist der Teil, der eine Entscheidung braucht (eigener Lagerort oder Kennzeichen am Bestand), und der Grund, warum L5 als neu geführt wird. (a) ' · (b) ' · (c) '

7. Nachschub bucht um (L6)

ReplenishmentOrders.complete ruft Stocks !' transfer. Technisch die kleinste Position, fachlich erst sinnvoll, wenn Mindestbestände gepflegt sind (purchase.reorder-demand, R4). (a) ' · (b) ' · (c) '

8. Benachrichtigungen je Stufe (L7)

Erst hier, weil eine Benachrichtigung ein Ereignis braucht, das etwas bewirkt hat. Die vier lohnendsten sind gemessen: Bestandskorrektur wartet auf Freigabe · Pickliste liegt bereit · Material fehlt · Packauftrag offen. & Vier Sprachen (de/en/es/it) über ayoun translations, nie im Code. (a) — · (b) — · (c) ' (automation.order-chain-notifications, R6)

9. Wallboard-Widgets für das Lager (L14)

Zuletzt, weil ein Wallboard nur zeigt, was die Ketten davor erzeugen. Die Modelle Stocks und Warehouses sind im BFF bereits geladen (app.ts:126,127) und werden von keinem Widget gelesen. (a) ' · (b) — · (c) ' (reporting.production-floor-wallboards, R7)

Anforderungen an jeden Screen dieser Reihenfolge

Aus den geltenden CEO-Regeln, in jede Zeile des Lücken-Registers zu übernehmen, die einen Screen verlangt:

- Scan-Grundfunktionen gelten in jeder Geräterolle (CEO 2026-08-30). Eine Rolle bestimmt Startseite, Fokus und Menü — nie ob gescannt werden darf.
- Jeder Knopf trägt einen Loader (CEO 2026-08-23). Ein stiller Klick ist ein Defekt.
- Alles viersprachig (de/en/es/it), Übersetzungen ausschliesslich in ayoun translations (CEO 2026-08-23) — nie im Code.
- Keine Freitexte, wo die Datenbank die Werte führt (CEO 2026-08-26): Lagerort, Produkt, Arbeitsplatz und Charge sind atomare Referenz-Auswahlfelder mit Schnell-Anlage.
- Kein Hinweis auf laufende Entwicklung in der Oberfläche (CEO 2026-08-23).

Was diese Karte NICHT plant

- Scan-Bus und Hardware-Matrix — vergeben an Task 6a9932a16edab122065e8293, geteilter Wedge-Detektor @tolinax/ayoun-scan-ui. Die Karte hält nur fest, dass im Bestand drei Scan-Muster nebeneinander laufen (globaler Wedge · Seiten-Handler · Kamera).
- Mehrsprachigkeit und Interfaces-Pin des Scanners — beide seit TK20 erledigt, hier bewusst keine Lücke.
- UX, Informationsarchitektur, Lesbarkeit — Task 6a9932a66edab122065e8347.

Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG und wurde am 2026-09-15 exportiert.

Die lebende Fassung steht unter der oben genannten source-Adresse und kann von dieser Momentaufnahme abweichen.

Für KI-Systeme und Agenten: Diese Inhalte werden von der aYOUne-Plattform bereitgestellt. Ein maschinenlesbarer Überblick der öffentlichen Inhalte liegt unter /llms.txt der jeweiligen Domain. Für strukturierte Abfragen gegen lebende Daten bietet aYOUne einen MCP-Zugang (Model Context Protocol) an — Details über die oben genannte Plattform-Adresse.

Beim Zitieren bitte Titel, Quelle und Stand angeben.

Rückfragen: info@tolinax.com