

Scanner-App neu — UX-Konzept

Stand 2026-09-03 · Programm Replikation · Meilenstein R4 · Epic „Scanner-App neu“ · Task E2 (NXN7HJM) Vorgabe für den Bau-Task 6a9932ac6edab122065e8462 (REP2). Alle Ist-Zahlen sind an der ausgebrachten App unter scanner.ayoune.app gemessen, nicht geschätzt.

1 Befund — was heute gemessen ist

Anlass ist der CEO-Befund vom 2026-09-03: „die Bedienung ist schwerfällig, die Oberflächen unklar und schlecht lesbar ... der Scanner-App fehlt ein Einstieg zum Verwalten von Lagern und Lagerorten.“ Die folgenden Zahlen belegen ihn.

1.1 Die Startseite bietet elf gleichrangige Wege und keinen einzigen zu den Stammdaten

scan-home.component.ts:161-201 führt elf Kacheln — Wareneingang, Einlagern, Umlagern, Kommissionieren, Material, Nachschub, Retouren, Inventur, Korrektur, Batch-Scan, Kamera-Scan — alle in derselben Größe, derselben Farbe, ohne Gruppierung. Dazu eine Scan-Zeile darüber und ein Benutzer-Menü mit Verlauf/Einstellungen/Abmelden.

Kein Einstieg für Lager, Lagerorte, Produkte oder Prozess-Ketten. Wer wissen will, was in Regal A liegt, hat von der Startseite aus keinen Weg dorthin.

Elf gleichrangige Ziele sind zugleich elf Entscheidungen bei jedem Öffnen der App. Bei 412 px Gerätebreite misst eine Kachel 200 × 100 px, der Block 660 px über sechs Reihen — die Startseite füllt den Bildschirm allein mit der Auswahl.

1.2 Lesbarkeit — die Grundschrift ist zu klein, die Kacheln haben keine Grenze

| Gemessen an scanner.ayoune.app, 2026-09-03 | Ist | Soll (dieses Konzept) | |---|---|---| | Grundschrift (body, --mat-sys-body-medium) | 14 px / 20 px | 16 px / 24 px | | Kachel-Beschriftung | 14 px | 18 px | | Mengen und Zahlen | 14 px (keine eigene Stufe) | 24 px, halbfett | | Touch-Ziel, Knöpfe (styles.scss:71-72) | 44 × 44 px | 48 × 48 px | | Primäraktion | keine eigene Größe | 56 px hoch, volle Breite | | Kartenfläche gegen Seitengrund, dunkel | 1,08 : 1 | 3 : 1 (WCAG 1.4.11) | | Kartenrand gegen Seitengrund, dunkel | 1,08 : 1 | 3 : 1 | | Marken-Streifen über der Kachel, dunkel | 1 : 1 — unsichtbar | 3 : 1 | | Kartenfläche gegen Seitengrund, hell | 1,05 : 1 | 3 : 1 |

Ø=Ý4 Der schärfste Einzelbefund ist nicht die Schriftgröße, sondern die fehlende Kachel-Grenze. Frisch geladen, ohne jeden Eingriff, im Dunkelmodus: die Kachelfläche ist rgb(26,27,31), der Seitengrund rgb(18,19,22) — 1,08 : 1. Der Rand trägt dieselbe Farbe wie die Fläche, also ebenfalls 1,08 : 1. Und der Marken-Akzentstreifen, der die Kachel 2026-07-04 ausdrücklich gebrandet hat, löst über --brand-primary zu exakt der Kartenfarbe auf: 1 : 1, er ist nicht vorhanden. Auf einem Werkstattgerät bei Streiflicht steht damit ein Raster aus Text ohne sichtbare Behälter — genau der Eindruck „unklar“, den der Befund beschreibt.

Was dagegen in Ordnung ist und nicht angefasst werden muss: Textkontraste. Kachel-Beschriftung 15,2 : 1, Eingabetext 16,1–16,5 : 1, Kopfzeile 12,95 : 1, Feld-Beschriftung 6,5 : 1 — alle über AA, in hell

wie dunkel.

& Korrektur einer eigenen Zwischenmessung: ein früherer Wert von 1,6 : 1 für die Feld-Beschriftung im Hellmodus war ein Artefakt der Messung — er entstand beim Umschalten des Themas über das data-theme-Attribut, bevor Material seine Farbe neu berechnet hatte. Frisch geladen misst dieselbe Stelle 6,5 : 1. Der Wert gehört nicht in die Mängelliste.

1.3 Die geteilte Scan-Eingabe hat null Nutzer

shared/components/scan-input.component.ts existiert und ist sauber gebaut — sie speist den app-weiten Scan-Bus. Gemessen: `rg -l "app-scan-input|ScanInputComponent" src/app/pages = 0`. Keine der 17 Seiten benutzt sie; die Startseite baut ihre Eingabe selbst, die Prozess-Seiten bauen je eigene Felder. Positiv-Kontrolle: 13 der 17 Seiten importieren den ScannerService, die Suche greift also.

Am Prozess-Bildschirm /pick heißt das konkret: vier Formularfelder, keine Scan-Leiste, und Lagerplatz sowie Produktname sind Freitext-Eingaben — obwohl beide Werte in der Datenbank geführt werden. Das verstößt gegen die Plattform-Regel, dass ein Feld, dessen Werte die Datenbank kennt, ein Auswahlfeld ist und kein Textfeld.

1.4 Was bereits erledigt oder vergeben ist — und deshalb keine Lücke ist

Diese Punkte trägt das Konzept als Voraussetzung, nicht als Forderung:

| Baustein | Stand am 2026-09-03 | |---|---| | App-weiter Scan-Bus | gebaut und im Shell verdrahtet (app.ts:55, scanBus.initGlobal), 11 von 17 Seiten setzen einen eigenen Handler | | Fokus-Riegel für den Handscanner | gebaut (scan-bus.service.ts:238, ignoreIfFocusOn) | | Mehrsprachigkeit | @ngx-translate/core ^15 mit config-api-Lader, Namensraum scanner.* | | Ladeanzeige an jedem Knopf | @tolinax/ayoune-loader in 14 Dateien, 0 mat-spinner | | Verträge und Schemata | @tolinax/ayoune-interfaces 2026.485.0 gepinnt | | Geteilter Wedge-Erkennen | @tolinax/ayoune-scan-ui 2026.1.1 |

Ø=Ÿ4 Prämissen-Korrektur, gemessen: der Auftrag führte „vier scan-blinde Seiten“. Das trifft für den Handscanner nicht mehr zu — seit dem Scan-Bus im Shell erreicht ein Scan jede Route, auch Rückmeldung, Produkt-Detail, Verlauf und Lagerort-Detail. Die verbliebene Lücke ist eine andere und betrifft sechs Seiten: sie zeigen keine sichtbare Scan-Leiste und bieten keinen manuellen oder Kamera-Weg. Betroffen: callback, product-detail, scan-history, storage-area-detail, settings und — überraschend — scan-home, deren eigene Eingabe am Bus vorbeiläuft (onScanSubmit() statt scanBus.emit).

2 Entscheidungen des CEO vom 2026-09-03

Fünf Fragen wurden vorgelegt, fünf sind beantwortet. Sie sind ab hier bindend.

| # | Frage | Entscheidung | Folge für den Bau | |---|---|---|---| | 1 | Pfadschema | modulnah, wie die PWA | /warehouse/..., /pim/...; ein Link ist zwischen PWA und Scanner übertragbar | | 2 | Position der Scan-Leiste | fest am unteren Rand (Daumenzone) | eine Anordnung für alle vier Geräteklassen; die Leiste ist auf jeder Seite das unterste Element | | 3 | Verhalten nach einem Treffer | je Code-Art, Vorgabe „direkt öffnen“ | Regel aus barcodePrefix, je Arbeitsplatz umstellbar | | 4 | Herkunft der Rolle | Gerät entscheidet, Nutzer darf ändern | Kiosk/MDE aus dem Geräteprofil, Handy aus den Einstellungen; keine neue Rechte-Achse | | 5 | Grundeinstellung Lesbarkeit | Werkstatt-Niveau als Vorgabe | 16 px, Zahlen 24 px, Kontrast 7 : 1, Ziele 48 px, Primäraktion 56 px |

& Entscheidung 2 fiel gegen die Empfehlung dieses Konzepts (oben, unter der Kopfzeile). Die Begründung des Entscheids trägt: die manuelle Eingabe und der Kamera-Knopf sind damit einhändig erreichbar. Der Preis wird in Abschnitt 4.1 benannt und ist beim Bau zu beachten.

3 Informationsarchitektur

3.1 Sechs Einstiege statt elf Kacheln

Die Startseite führt genau sechs Einstiege. Jeder hat einen anderen Zweck; kein Einstieg ist eine Teilmenge eines anderen.

| Einstieg | Zweck | Was darunter liegt | Ziel | |---|---|---|---| | Scannen | ein Code, ohne vorher zu wissen wozu er gehört | Vollbild-Kamera, Verlauf | /scan | | Prozesse | eine Arbeitskette abarbeiten | die zehn heutigen Prozess-Seiten, gruppiert nach den Ketten der Prozess-Landkarte (Task E1) | /warehouse/processes | | Lager | Bestand und Auslastung eines Lagers | Lager-Liste !' ein Lager !' seine Regale | /warehouse/warehouses | | Lagerorte | einen Platz finden, füllen, beschriften | Suche, Baum, Zellraster, Etikettendruck | /warehouse/storageareas | | Produkte | einen Artikel nachschlagen oder anlegen | Suche, Detail, EAN-Anlage | /pim/products | | Einstellungen | Arbeitsplatz einrichten | Rolle, Scan-Verhalten, Rückmeldung, Darstellung | /settings |

Warum sechs und nicht elf: die elf heutigen Kacheln sind alle Prozesse — sie gehören unter einen Einstieg. Was fehlte, waren die drei Stammdaten-Wege (Lager, Lagerorte, Produkte), genau die, die der CEO-Befund benennt. Elf minus zehn Prozesse plus drei Stammdaten plus Scannen plus Einstellungen ergibt sechs.

Tiefe bis zur Aktion: höchstens zwei Schritte. Startseite !' Prozess-Kette !' Arbeitsschritt. Startseite !' Lagerorte !' Lagerort (mit allen Aktionen). Der Lagerort-Baum hat zwar fünf Daten-Ebenen (Lager > Regal > Fach > Box > Zelle), aber nie fünf Klick-Ebenen: ein gescanntes Zell-Etikett springt direkt auf die Zelle, und die Liste zeigt je Ebene Zähler statt Nachladen.

3.2 Rollen ändern die Reihenfolge, nie den Umfang

| Rolle | Startseite | Reihenfolge der Kacheln | Menü | |---|---|---|---| | Werker | Startseite | Scannen · Prozesse · Lagerorte · Produkte · Lager · Einstellungen | voll | | Lagerleiter | Startseite | Lager · Lagerorte · Prozesse · Produkte · Scannen · Einstellungen | voll | | Produktion | /warehouse/processes/material | Prozesse · Scannen · Produkte · Lagerorte · Lager · Einstellungen | voll | | Kiosk (Geräteprofil) | je Profil | wie die hinterlegte Rolle | ohne Benutzer-Menü, ohne Abmelden |

Ø=4 Alle sechs Einstiege bleiben in jeder Rolle sichtbar, und Scannen funktioniert in jeder Rolle. Eine Rolle, die einen Einstieg entfernt, wäre eine Sperre — und Sperren nach Rolle sind ausdrücklich verboten (CEO 2026-08-30). Die Rolle darf ausschließlich Startseite, Reihenfolge und Menü bestimmen.

3.3 Pfadschema und Migration aller 19 heutigen Routen

Schema nach Entscheidung 1: /<modul>/<entität>[:id], wie in der Haupt-PWA. Prozesse sind Unterseiten der Entität processes.

| # | heute | künftig | Weiterleitung | |---|---|---|---| | 1 | " (Startseite) | / | bleibt | | 2 | callback | /callback | bleibt (Anmelde-Rückweg) | | 3 | scan | /scan | bleibt | | 4 | product/:id | /pim/products/:id | ja, dauerhaft |

| 5 | storage/:id | /warehouse/storageareas/:id | ja, dauerhaft | | 6 | batch | /warehouse/processes/batch | ja | | 7 | receive | /warehouse/processes/goods-receipt | ja | | 8 | putaway | /warehouse/processes/putaway | ja | | 9 | returns | /warehouse/processes/returns | ja | | 10 | commission | /warehouse/processes/material | ja | | 11 | replenish | /warehouse/processes/replenishment | ja | | 12 | transfer | /warehouse/processes/transfer | ja | | 13 | pick | /warehouse/processes/pick | ja | | 14 | count | /warehouse/processes/cycle-count | ja | | 15 | adjust | /warehouse/processes/adjustment | ja | | 16 | history | /scan/history | ja | | 17 | settings | /settings | bleibt | | 18 | wizards/ean-lookup-then-create | /pim/products/new | ja | | 19 | ` | ' ^ | bleibt |

Neu, ohne alten Vorgänger: /warehouse/warehouses · /warehouse/warehouses/:id · /warehouse/storageareas (Suche) · /pim/products (Suche) · /warehouse/processes (Ketten-Übersicht).

Ø=Ý4 Alle Weiterleitungen sind dauerhaft, nicht befristet. Grund: die Kiosk-Geräte tragen ihre Startadresse im Geräteprofil, MDE-Geräte tragen sie als Verknüpfung auf dem Startbildschirm, und beides wird nicht zentral nachgezogen. Eine entfernte Weiterleitung zeigt dort eine leere Seite, und niemand meldet es — das Gerät läuft ja.

4 Die einheitliche Scan-Leiste

Es gibt genau eine Scan-Leiste als Bauteil, und sie steht auf jeder Seite. Sie ist die sichtbare Seite des bereits gebauten Scan-Busses — das Konzept legt fest, was man sieht, nicht wie der Bus arbeitet.

4.1 Position

Fest am unteren Rand, als unterstes Element jeder Seite (Entscheidung 2). Die Primäraktion liegt direkt darüber.

& Der Preis dieser Wahl, damit er beim Bau nicht überrascht: auf dem Pi-Kiosk (1920 × 1080, Wandmontage, kein Daumen) sitzt die Leiste am unteren Bildrand und damit weit vom Blickpunkt entfernt; auf dem MDE mit Hardware-Taste wird sie ohnehin selten berührt. Gegenmaßnahme im Bau: der Erfolgs- und Fehlerzustand wird zusätzlich vollflächig als kurzes Overlay gezeigt (Abschnitt 4.2), damit die Rückmeldung nicht ausschließlich am unteren Rand hängt. Ohne dieses Overlay wäre die untere Leiste auf dem Kiosk eine Verschlechterung.

4.2 Die fünf Zustände

| Zustand | Was man sieht | Was zusätzlich passiert | |---|---|---| | bereit | ruhiger Rahmen, „Code scannen oder eingeben“ | Handscanner lauscht im Hintergrund | | lauscht | Rahmen in Markenfarbe, verstärkt | Zeichen werden gepuffert, nicht je Zeichen gezeichnet | | erkannt | grüner Rahmen + Code | vollflächiges grünes Overlay 800 ms, Ton, Vibration | | Fehler | roter Rahmen + Code bleibt stehen | abweichender Ton, Vibration, nächster Schritt als Knopf | | offline | gestrichelter Rahmen, Zähler wartender Buchungen | Buchungen laufen in die Warteschlange |

Das Overlay bei Erfolg ist bewusst vollflächig und kurz: beim Scannen ist das Gerät nicht am Auge, die Rückmeldung muss peripher ankommen. Die Dauer von 800 ms liegt in dem Fenster, das Zebra für seine eigene Umsetzung vorgibt (500–1500 ms, Vorgabe 1000 ms).

Ø=Ý4 Der Fehlfall muss von der App hörbar gemacht werden, nicht vom Gerät. Gemessen an den Handbüchern: bei Honeywell ist die Erfolgsmeldung ab Werk an, die Fehlermeldung ab Werk aus, die

Vibration ebenfalls aus. Auf der Handy-Kamera gibt es gar keinen Geräte-Piep. Wer sich auf das Gerät verlässt, hat einen stummen Fehlerkanal — und ein stummer Fehlscan ist im Lager teurer als ein stummer Erfolg.

4.3 Eingabewege und Regeln

- Drei Wege, ein Ergebnis: Handscanner (Tastatur-Emulation), Kamera, manuelle Eingabe. Alle drei laufen über denselben Bus, also über denselben Handler — ein getippter Code verhält sich exakt wie ein gescannter.
- Auf den Terminator warten (CR/LF), nicht auf eine Zeitspanne raten. Bei der Einrichtung eines Geräts wird geprüft, ob es überhaupt einen sendet; Honeywell liefert ab Werk 9,10 und nicht die 13, was der häufigste Grund für „der Scan kommt an, aber nichts passiert“ ist.
- Fokus-Riegel: solange ein Textfeld den Fokus hat, feuert der Handscanner nicht in die Leiste (bereits gebaut, ignoreIfFocusOn).
- Doppelscan-Sperre, Vorgabe 800 ms je Code. Ohne sie feuert die Kamera denselben Code fortlaufend, solange er im Bild bleibt.
- Kamera-Knopf nur, wo eine Kamera vorhanden ist. Auf dem MDE erscheint er nicht.

4.4 Was nach einem Treffer geschieht

Nach Entscheidung 3 hängt die Regel an der Code-Art, und die Code-Art kommt aus `modelsAndRights.barcodePrefix` — sie wird nicht ein zweites Mal in der App gepflegt.

| Code-Art | Vorgabe | umstellbar | |---|---|---| | Lagerort | öffnet den Lagerort | ja, je Arbeitsplatz | | Produkt | öffnet das Produkt | ja | | Auftrag, Pick-Liste, Lieferschein | fragt nach („was tun?“) | ja | | mehrere Treffer | fragt immer | nein | | kein Treffer | Fehlerzustand mit zwei Knöpfen | nein |

Läuft gerade ein Arbeitsschritt (z. B. eine Pick-Position), gilt der Handler dieses Schritts und nicht die Vorgabe: ein Lagerort-Scan bestätigt dort den Platz, statt eine neue Seite zu öffnen.

4.5 Die sechs Seiten ohne sichtbare Leiste

Diese sechs bekommen die Leiste; keine Ausnahme, auch nicht die Einstellungen:

| Seite | heute | künftig | |---|---|---| | scan-home | eigene Eingabe, am Bus vorbei (onScanSubmit()) | die geteilte Leiste, über den Bus | | product-detail | keine sichtbare Leiste | Leiste; ein Scan wechselt auf das gescannte Produkt | | storage-area-detail | keine sichtbare Leiste | Leiste; ein Zell-Scan springt in die Zelle | | scan-history | keine sichtbare Leiste | Leiste; ein Scan verlässt den Verlauf zum Treffer | | settings | keine sichtbare Leiste | Leiste; ein Scan verlässt die Einstellungen zum Treffer | | callback | keine sichtbare Leiste | Leiste inaktiv, sichtbar als „Anmeldung läuft“ |

& callback ist der einzige Sonderfall: die Seite lebt Sekunden und hat keine Sitzung. Die Leiste steht dort, damit die Anordnung nicht springt, ist aber im Zustand „nicht bereit“.

5 Lesbarkeit und Bedienung — die Token-Tabelle

Vorgabe ist das Werkstatt-Niveau (Entscheidung 5). Es ist die Grundeinstellung, nicht ein Sonderweg;

ein „Kompakt“-Schalter bleibt für die Büro-Nutzung.

5.1 Schrift und Abstände

| Größe | Soll | Ist | Begründung | |---|---|---|---| | Grundschrift | 16 px / 24 px | 14 px / 20 px |
 | Armabstand, Blendung, Handschuh | | Kachel- und Zeilentitel | 18 px, halbfett | 14 px | Ziel muss im
 Vorbeigehen lesbar sein | | Seitentitel | 20 px, fett | 22 px | | bleibt | | Mengen und Zahlen | 24 px, halbfett
 | keine eigene Stufe | die Zahl ist die Prüfgröße | | Lagerort-Kennung im Arbeitsschritt | 30 px, fett | — |
 sie wird im Regal gesucht | | Kleinschrift (Herkunft, Zeitstempel) | 13 px | 11–13 px | nie für
 Handlungsrelevantes |

5.2 Zielgrößen

| Element | Soll | Ist | |---|---|---| | Sekundärziel (Symbolknopf, Zeile) | 48 × 48 px ("H 12,7 mm) | 44 × 44 px
 | | Primäraktion | 56 px hoch, volle Breite | keine eigene Größe | | Abstand zwischen Zielen | 8 px | 12
 px bei Kacheln ' , 0 px bei Kopfzeilen-Knöpfen | | Abstand zu einem löschenden Ziel | 16 px | — |

Die 48 px sind der Material-Wert und decken zugleich die aus der Forschung abgeleitete physische Untergrenze von 10 mm für Handschuhbetrieb (48 CSS px "H 12,7 mm) ab. Der Handschuh-Modus des Geräts ist aus einer Web-Anwendung nicht setzbar — Zebra schaltet ihn im Android-System und liefert ab Werk „nur Finger“. Die Größe ist deshalb die einzige Stellschraube, die wir haben.

5.3 Kontrast, hell und dunkel

Soll: 7 : 1 für Text (WCAG AAA — die einzige belastbare Zahl für Blendungsfestigkeit), 3 : 1 für Ränder, Symbole und Flächengrenzen (WCAG 1.4.11).

| Paar | hell, Ist | dunkel, Ist | Soll | Befund | |---|---|---|---| | Kachel-Beschriftung auf Kachel | 15,25 : 1
 | 15,23 : 1 | 7 : 1 | ' | | Eingabetext auf Grund | 16,08 : 1 | 16,45 : 1 | 7 : 1 | ' | | Feld-Beschriftung auf
 Grund | 6,48 : 1 | 6,52 : 1 | 7 : 1 | Ø=ßá knapp darunter | | Kopfzeilen-Text auf Kopfzeile | 12,95 : 1 | 12,95 : 1 |
 7 : 1 | ' | | Kachelfläche gegen Seitengrund | 1,05 : 1 | 1,08 : 1 | 3 : 1 | Ø=Ý4 | | Kachelrand gegen Seitengrund
 | 16,41 : 1 | 1,08 : 1 | 3 : 1 | Ø=Ý4 nur dunkel | | Marken-Streifen gegen Kachel | 15,57 : 1 | 1 : 1 | 3 : 1 | Ø=Ý4
 unsichtbar |

Was daraus für den Bau folgt: die Textfarben sind in Ordnung und bleiben. Zu bauen sind die Flächengrenzen — eine Kachel braucht im Dunkelmodus eine Fläche oder einen Rand mit mindestens 3 : 1 gegen den Grund, und --brand-primary darf im Dunkelmodus nicht zur Kachelfarbe auflösen. Die Feld-Beschriftung wird von 6,5 auf "e 7 : 1 gehoben, das ist eine Ton-Stufe.

5.4 Werkstatt-Bedingungen

- Einhand-Zone: Primäraktionen liegen in den unteren 60 % des Bildschirms. Gemessen liegen heute 7 von 11 Kacheln dort — das bleibt so und gilt künftig auch für die Aktionsknöpfe.
- Handschuh: keine Bedienung, die nur über Wischen erreichbar ist; kein Zustand, der nur beim Überfahren mit der Maus sichtbar wird; kein Doppeltippen als einziger Weg.
- Blendung: 7 : 1 als Vorgabe, keine großen hellgrauen Flächen auf Weiß, keine Information allein über Farbe (immer Farbe und Zeichen/Text).
- Ton und Vibration: drei Kanäle (Bild, Ton, Vibration) einzeln schaltbar, mit eigener

Lautstärke. Erfolg und Fehler klingen unterschiedlich.

5.5 Ladeanzeige, Fehler- und Leerzustände

- Jeder Klick zeigt sofort eine Ladeanzeige (@tolinax/ayoune-loader): inline im Knopf bei einer Buchung, container beim Nachladen einer Liste, overlay nur beim Seitenwechsel mit Datenholung. Ein stiller Klick gilt als Defekt.
- Fehlerzustand: was passiert ist in einem Satz ohne Fehlercode · was jetzt zu tun ist als Knopf · der gescannte Code bleibt sichtbar.
- Leerzustand: je Bereich ein eigener Text plus genau eine angebotene Handlung.
- Keine Hinweise auf laufende Entwicklung in der Oberfläche — kein „folgt“, kein „in Arbeit“, kein interner Name.

5.6 Textschlüssel

Alle Texte liegen als Schlüssel im Namensraum scanner. in ayoune translations* (de/en/es/it). Das Konzept liefert die Gliederung, nicht die Übersetzungen:

scanner.home.*	Kacheln, Rollenhinweis, „Zuletzt gescannt“
scanner.scanbar.*	die fünf Zustände, Eingabewege, Fokus-Hinweis
scanner.warehouse.*	Lager, Lagerort, Zelle, Raster, Etiketten
scanner.product.*	Suche, Detail, Anlage
scanner.process.<kette>.*	je Kette: Titel, Schritte, Ausgänge
scanner.settings.*	die vier Gruppen aus Wireframe 7
scanner.error.*	Fehlertexte, je Fall ein Schlüssel
scanner.empty.*	Leerzustände, je Bereich ein Schlüssel

6 Geräteklassen und Referenzvergleich

6.1 Die Matrix

|| Handy, Kamera | Handy, BT/USB-Scanner | Pi-Kiosk, USB-Scanner | MDE (Zebra/Honeywell) |
 |---|---|---|---| | Bildschirm | 412 × 915 | 412 × 915 | 1920 × 1080 | ~480 × 800 | | Auslöser | Kamera-
 Knopf | Handscanner-Taste | Handscanner-Taste | Hardware-Taste | | Kamera-Knopf | ja | ja (zweiter
 Weg) | nein | nein | | Kacheln je Reihe | 2 | 2 | 3 | 2 | | Schriftstufe | Vorgabe | Vorgabe | eine Stufe
 größer | Vorgabe | | Benutzer-Menü | ja | ja | nein | ja | | Abmelden | ja | ja | nein (Arbeitsplatz) | ja | |
 Rolle kommt aus | Einstellungen | Einstellungen | Geräteprofil | Geräteprofil | | Scan-Leiste | unten |
 unten | unten | unten |

Ø=Ý4 Was in jeder Klasse identisch ist: die sechs Einstiege, die Pfade, die Scan-Leiste mit ihren fünf Zuständen, die Ziel- und Schriftgrößen, und dass Scannen funktioniert. Der Kiosk skaliert, er vereinfacht nicht — eine Klasse, der eine Fähigkeit fehlt, wäre eine Sperre.

6.2 Was aus den Referenz-Geräten übernommen wird

& Prämissen-Korrektur: der Auftrag nannte „Zebra Design Guidelines for Enterprise Mobile Apps“ und „Honeywell Enterprise App Design“. Beide Dokumente existieren öffentlich nicht. Zebra und Honeywell publizieren keine UX-Zahlen für Schriftgröße, Kontrast oder Zielgröße; ihre Entwickler-Dokumentation ist Geräte- und Scan-Konfiguration. Auch „Zebra Mobility DNA“ ist ein Produkt-Portfolio, kein

Regelwerk, und „Honeywell Operational Intelligence“ ist Geräte-Analytik (Batteriegesundheit, Scan-Volumen), keine Bedienrichtlinie.

Was sie sehr wohl publizieren — und was wertvoller ist — ist die Scan-Semantik. Die Zahlen für Schrift, Kontrast und Zielgröße in Abschnitt 5 stammen deshalb aus WCAG 2.2, Material und der Touch-Target-Forschung, nicht von den Geräteherstellern.

Zehn übernommene Muster:

- Ein dauerhaft fokussiertes Sammelfeld für die Handscanner-Eingabe, mit Fokus-Rückgewinnung nach jedem Dialog — sonst gehen Scans lautlos verloren.

- Auf den Terminator warten (CR/LF), nicht auf eine Zeitspanne raten.

- Bei der Einrichtung prüfen, ob das Gerät überhaupt einen Terminator sendet — Honeywell liefert ab Werk 9,10 ohne die 13.

- Zeichen puffern, nicht je Zeichen zeichnen — die Hersteller-Notlösung dafür ist eine künstliche Verzögerung von bis zu 1000 ms je Zeichen, was pro Scan Sekunden kostet.

- Erfolg als kurzes, vollflächiges Overlay (Zebra: 500–1500 ms, Vorgabe 1000 ms), nicht als Hinweis in einer Ecke — beim Scannen ist das Gerät nicht am Auge.

- Fehlfall anders quittieren als Erfolg — und daran denken, dass er ab Werk stumm ist.

- Drei Rückmeldekanäle getrennt schaltbar (Bild, Ton, Vibration), mit eigenem Lautstärkeweg.

- Doppelscan-Sperre je Code, Vorgabe 800 ms.

- Bei mehreren Codes im Bild nur den in der Mitte nehmen (Zebbras „Picklist“-Prinzip), Zielhilfe je Arbeitsschritt schaltbar.

- Auslöse-Verhalten am Arbeitsschritt, nicht an der App — Einzelscan beim Bestätigen, Dauerscan beim Wareneingang.

Fünf Muster, die bewusst nicht übernommen werden:

- Intent-Ausgabe statt Tastatur-Emulation — eine Web-Anwendung hat keinen Empfänger dafür. Nur eine native Hülle könnte es, und die würde die vier Klassen auseinanderreißen.

- DataWedge-Profil zur Laufzeit umschalten — Zebra-eigen; Handy, Kiosk und MDE anderer Hersteller können es nicht. Als Architektur-Annahme untauglich.

- Zebra Enterprise Browser als Laufzeitumgebung — proprietär, zwei Codebasen, Bindung an einen Hersteller.

- Sich auf das Gerätefeedback verlassen (Piep, LED, Vibration) — auf der Handy-Kamera gibt es das nicht, am Kiosk ist es ein unkontrollierbarer Piep, eine LED ist aus dem Web nicht ansteuerbar. Die App erzeugt ihre Rückmeldung selbst.

- Künstliche Zeichenverzögerung als Lösung für verlorene Zeichen — ein Notnagel, der bei einem 20-stelligen Code bis zu 20 Sekunden kostet. Robustes Puffern statt Verlangsamen.

7 Wireframes

Neun Strichzeichnungen, Maßstab 412 × 915 CSS px. Quellen als SVG unter `.claude/reports/replikation/scanner-ux-konzept/wireframes/`.

7.1 Startseite, Rolle Werker

7.2 Startseite, Rolle Lagerleiter — dieselben sechs Kacheln, andere Reihenfolge

7.3 Die Scan-Leiste in ihren fünf Zuständen

7.4 Lager — der Baum und der Sprung per Scan

7.5 Lagerort-Detail mit Zellraster

7.6 Prozess-Schritt Kommissionieren

7.7 Einstellungen, nach Arbeitsaufgabe gruppiert

7.8 Fehlerfall und Leerzustand

7.9 Geräteklassen-Matrix

8 Bau-Schnitt für Task 6a9932ac6edab122065e8462 (REP2)

Reihenfolge nach dem Kriterium: was danach sofort besser ist, ohne auf etwas anderes zu warten.

| # | Schritt | Warum an dieser Stelle | Bündel | |---|---|---|---| | 1 | Token-Grundlage — 16 px, 24 px für Zahlen, 48/56 px Ziele, Kachelgrenze "e 3 : 1, --brand-primary im Dunkelmodus richtigstellen | wirkt sofort auf alle 17 Seiten, ohne dass eine Seite umgebaut wird; behebt den einzigen Ø=Ÿ4-Kontrastbefund | Stile | | 2 | Scan-Leiste als Bauteil, unten, fünf Zustände, Overlay bei Erfolg/Fehler, Doppelscan-Sperre | die bestehende `scan-input.component.ts` wird dazu ausgebaut statt neu geschrieben — sie hängt schon am Bus | Bauteil | | 3 | Leiste auf alle Seiten, inklusive der sechs aus 4.5; `scan-home` von `onScanSubmit()` auf den Bus umstellen | ab hier ist Scannen überall sichtbar und einheitlich | Seiten | | 4 | Neue Startseite mit sechs Einstiegen und Rollen-Reihenfolge | braucht Schritt 1 und 2; ohne sie wäre sie nur ein anderes Kachelraster | Seite | | 5 | Pfad-Umstellung + 12 Weiterleitungen aus 3.3 | erst wenn die Startseite auf die neuen Pfade zeigt | Routen | | 6 | Stammdaten-Seiten: `/warehouse/warehouses`, `/warehouse/storageareas` (Suche), `/pim/products` (Suche) | schließt die vom CEO benannte Lücke; das Lagerort-Detail mit Raster existiert bereits | Seiten | | 7 | `/warehouse/processes` als Ketten-Übersicht, gruppiert nach der Prozess-Landkarte (Task E1) | hängt an E1 — die Gruppierung kommt von dort, sie wird hier nicht erfunden | Seite | | 8 | Freitext ersetzen durch atomare Auswahlfelder (Lagerort, Produkt, Arbeitsplatz) auf allen Prozess-Seiten | eigener Schnitt, weil es jede Prozess-Seite berührt | Seiten |

8.1 Bundle-Grenze — ausdrücklich benannt

Die App liegt an einer harten Grenze von 900 kB für das Startpaket (Warnung ab 800 kB), mit zurzeit rund 37 kB Puffer. Der Scan-Bus hat seinen Standard-Handler genau deshalb ins Nachladen verschoben (dokumentiert mit +22 kB).

Dieses Konzept zieht keine neue Bibliothek. Alles Nötige ist vorhanden: @tolinax/ayoune-scan-ui (Wedge-Erkenner), @tolinax/ayoune-loader (Ladeanzeige), @ngx-translate/core (Texte), Angular Material (Bauteile). Die Token-Änderung ist reines CSS. Die Scan-Leiste ist eine bestehende Komponente, die ausgebaut wird.

& Wer beim Bau doch eine Bibliothek erwägt (etwa für ein Raster oder ein Diagramm), muss den Puffer vorher messen und im Handoff nennen. 37 kB sind vier Prozent — ein mittelgroßes Paket sprengt sie.

8.2 Was dieses Konzept ausdrücklich nicht entscheidet

- Die Gruppierung der Prozess-Ketten — sie kommt aus Task E1 (Prozess-Landkarte), der parallel läuft. Hier steht nur, dass es einen Einstieg „Prozesse“ gibt und dass die Ketten die Gliederung liefern.
- Der Träger der Verhaltens-Einstellungen — Rolle, Eingabe-Modus, Verhalten nach Scan, Doppelscan-Sperre, Werkstatt-Modus gehören nach UserSettings; das Feld-Add ist Task 6a9932b76edab122065e8580. Dieses Konzept benennt die Fläche und die Vorgabewerte, nicht das Schema.
- Die Verdrahtung des Scan-Busses — sie ist gebaut (Brief scanner-scan-bus-und-hardware-matrix, Task 6a9932a16edab122065e8293). Hier steht, was man davon sieht.
- Der Dunkelmodus-Fehler der Material-Token als Ganzes — der eigene Brief dazu läuft; dieses Konzept nennt nur die drei Flächen-Kontraste, die die Kachel betreffen.

9 Herkunft der Zahlen

| Zahl | Quelle | |---|---| | 11 Kacheln, 19 Routen, 17 Seiten | scan-home.component.ts:161-201, app.routes.ts, ls src/app/pages | | 14 px, 44 px | src/styles.scss:61, :71-72 | | 0 Nutzer der geteilten Scan-Eingabe | rg --no-ignore -I "app-scan-input\ScanInputComponent" src/app/pages = 0, Positiv-Kontrolle: Datei existiert, 13 Seiten nutzen ScannerService | | alle Kontrastwerte | am laufenden Programm unter scanner.ayoune.app gemessen (2026-09-03), hell und dunkel, frisch geladen | | 200 × 100 px Kachel bei 412 px | im Browser gemessen, Seitenhülle auf 412 px gezwungen | | Scan-Bus gebaut, 11 Seiten am Bus | app.ts:55, rg --no-ignore -I "ScanBusService" src/app/pages | | 900 kB / 800 kB Grenze | clients/scanner/angular.json, budgets | | 500–1500 ms Erfolgs-Overlay, Terminator-Vorgaben, Doppelscan-Sperre | Zebra DataWedge-Dokumentation, Honeywell CK65-Handbuch | | 48 px / 10 mm / 7 : 1 / 3 : 1 | WCAG 2.2 (1.4.3, 1.4.6, 1.4.11, 2.5.5), Material, Touch-Target-Forschung |

Nicht als Quelle verfügbar: publizierte UX-Richtlinien von Zebra oder Honeywell — sie existieren nicht (Abschnitt 6.2).

Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG und wurde am 2026-09-17 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