

Tech Stack (CEO approved 2026-03-03)

Tech Stack (CEO approved 2026-03-03)

Decision Date

CEO Approval: 2026-03-03 **Supersedes:** Express + TypeScript backend, monolithic Next.js frontend

Backend

Component	Choice	Rationale
Language	Kotlin	Type-safe, JVM performance, JetBrains ecosystem
Framework	Ktor	Coroutine-based async, lightweight, idiomatic Kotlin
Database	PostgreSQL 16	Rock-solid relational DB, JSONB for flexible fields
ORM	Kotlin Exposed	Type-safe SQL DSL, no annotation magic
Build tool	Gradle (Kotlin DSL)	Standard for Kotlin/JVM projects
Testing	Kotest + Testcontainers	BDD-style tests, real DB for integration tests
Auth	JWT (24h expiry) + refresh tokens	Standard stateless auth
Queue	Kotlin coroutines + Redis	Async jobs: label generation, carrier calls

Frontend — Micro-Frontend Architecture

Component	Choice	Rationale
Framework	React 19 + TypeScript	Stable, large ecosystem
Bundler	Vite + Module Federation	MFE-ready, fast HMR
Architecture	Micro-Frontend (MFE)	Each feature independently deployable
Shared UI	Tailwind CSS 4 + shadcn/ui	Design tokens in packages/ui/
State	Zustand per MFE	Lightweight, no global store coupling

MFE Structure

MFE	Responsibility	Deploy
shell	Layout, routing, auth, navigation	Always
mfe-inventory	Inventory management, bin locations, cycle counts	Independent
mfe-orders	Order management, inbound/outbound	Independent
mfe-picking	Picking workflows, wave management	Independent
mfe-settings	User management, integrations, config	Independent
mfe-ai	Plock AI Chat, analytics, dashboards	Independent

Each MFE is **independently deployable** — a change to mfe-picking does not require redeploying the shell or other MFEs.

AI / ML Layer

Component	Choice	Use Case
LLM	Claude API (claude-sonnet-4-5)	Plock AI Chat — Swedish NL queries
ML framework	Python microservice	Isolation from JVM, flexible ML tooling
Route optimisation	OR-Tools (Google)	Smart Picking — VRP/TSP solver
Anomaly detection	scikit-learn	Z-score + isolation forest

Component	Choice	Use Case
Forecasting (Phase 2)	XGBoost / Prophet	Demand forecasting

Infrastructure

Component	Choice
Monorepo	Turborepo (frontend MFEs) + Gradle (Kotlin backend)
Mobile (Phase 2)	React Native — scanner + picking app
Cache	Redis
Testing (frontend)	Vitest + Playwright
CI/CD	GitHub Actions
Containers	Docker + Docker Compose (local dev)

Architecture Decision Records

ADR-001: Kotlin over TypeScript for backend (2026-03-03)

Context: Previous design used Express + TypeScript. Decision to move to Kotlin + Ktor. Rationale: JVM performance for WMS workloads, stronger type safety with Exposed ORM, coroutine-based concurrency fits async warehouse operations (carrier calls, label generation). Status: Accepted by CEO 2026-03-03.

ADR-002: Micro-Frontend over monolithic Next.js (2026-03-03)

Context: WMS has distinct feature domains that evolve at different rates (picking vs AI vs integrations). Rationale: Independent deployment reduces risk, enables feature teams to work in parallel, allows phased feature releases. Trade-off: Higher initial complexity, but pays off at scale. Status: Accepted by CEO 2026-03-03.

Revision #3

Created 2026-03-04 05:08:51 UTC by John

Updated 2026-05-31 20:04:50 UTC by John