

Canonical Source of Truth

Source: `~/ALAI/products/Plock/docs/demo-readiness/00-canonical-source-of-truth.md`

PLOCK — Canonical Source of Truth

Date: 2026-03-14

Purpose: Define which files are authoritative for PLOCK and which files are stale/conflicting, so all future documentation, QA, and implementation work is based on real system state.

1. Why This Exists

PLOCK currently has a real codebase and substantial documentation, but not all documentation reflects the same architecture.

Without a canonical source-of-truth map:

- user stories can be extracted from stale assumptions
- feature catalogs can be built against the wrong architecture
- QA can validate one document against another document that is itself outdated
- implementation tasks can be generated against a system that does not match the actual repo

This document defines what is authoritative and what is not.

2. Authority Order

When files conflict, use this precedence:

1. **Actual code and repository structure**
2. **Database migrations and runtime configuration**
3. **BUILD-BLUEPRINT.md**

4. **README.md**
 5. **Runbooks and operational docs**
 6. **ADR documents**
 7. **PRD / GTM / Regulatory docs**
 8. **API-SPEC.md** (reference only unless confirmed against code)
 9. **Stale/conflicting docs must not be used as source of truth**
-

3. Authoritative Sources

3.1 Codebase

The following repo areas are authoritative:

- backend/
- frontend/
- backend/src/main/resources/db/migration/
- tests/
- e2e/
- infrastructure/

3.2 Primary Architecture / Build Docs

These are currently the strongest documentation sources:

- README.md
- BUILD-BLUEPRINT.md
- RUNBOOK.md
- docs/RUNBOOK.md
- docs/RLS-AUDIT.md
- docs/SECRETS-MANAGEMENT.md

3.3 Product / Business / Compliance Docs

These are authoritative for product intent, market framing, compliance assumptions, and user-level needs — but must still be reconciled against code:

- docs/PRD.md
- docs/GTM-STRATEGY.md
- docs/REGULATORY.md

3.4 ADRs

Use as authoritative for declared decisions when they match repo reality:

- `adr/ADR-001-flyway-over-prisma.md`
 - `adr/ADR-002-module-federation.md`
 - `adr/ADR-003-rls-tenant-isolation.md`
-

4. Current Real Architecture

Based on actual code and authoritative docs, the real current system is:

Backend

- **Kotlin**
- **Ktor**
- **Exposed ORM**
- **Flyway migrations**
- **PostgreSQL**
- **Redis**

Frontend

- **React 19**
- **TypeScript**
- **Vite**
- **Module Federation**
- **Micro-frontends**
 - `shell`
 - `mfe-inventory`
 - `mfe-orders`
 - `mfe-picking`
 - `mfe-settings`
 - `mfe-ai`

Infrastructure / Deployment

- Railway-oriented deployment and infra
- Docker / docker-compose for local setup
- Terraform and deployment helper scripts under `infrastructure/`

Multi-tenancy

- Tenant boundary is `warehouse_id`
- Isolation enforced via **PostgreSQL RLS**
- Tenant context set through application request flow

Integrations

- Fortnox
 - PostNord
 - DHL
 - Instabee
-

5. Semi-Authoritative / Needs Verification

`docs/API - SPEC .md`

This file is useful as a product/API intent document, but it must be verified against the real Ktor routing and actual backend implementation before being treated as canonical.

Use it as:

- reference
- gap analysis input
- API-intent documentation

Do **not** use it as final truth without code verification.

6. Stale / Conflicting Sources

`docs/ARCHITECTURE .md`

This file is currently **stale/conflicting** and must **not** be used as the main architecture source.

It describes a different architecture, including:

- Next.js
- Express
- Prisma
- Python ML microservice
- Supabase
- Upstash
- `organization_id`-based tenancy

That conflicts with the actual repo and stronger sources, which show:

- Kotlin/Ktor
- Exposed + Flyway
- React + Vite MFEs
- PostgreSQL + Redis
- `warehouse_id` tenant model

Rule: `docs/ARCHITECTURE.md` may be used only as historical/stale material until rewritten or archived.

7. Operational Constraints

7.1 Agent Output Is Not Source of Truth

The following are **not authoritative**:

- `/tmp/verify-*`
- `/tmp/gotcha-task-*`
- agent-generated summaries that claim files were created but no files exist
- MC task status alone (`done`, `blocked`, etc.) without artifact verification

7.2 Deliverable Existence Is Required

Any documentation task that claims output files must be verified against the actual filesystem. A claimed file path is not considered real unless:

- the file exists
 - it is readable
 - its contents are relevant to the requested task
-

8. What Future Docs Must Be Based On

The next canonical PLOCK documentation pass must be based on:

User Stories

- `docs/PRD.md`
- actual route/service domains in backend
- AI strategy / GTM / regulatory documents where relevant

Feature Catalog

- backend routes and services
- frontend MFEs
- migrations / schema
- integrations
- security / RLS docs

ADR / BDR

- `adr/*`
- BUILD-BLUEPRINT
- actual code decisions already visible in repo
- business decisions visible in PRD / GTM / pricing docs

Test Scenarios

- existing backend tests
 - existing e2e tests
 - route/service/module structure
 - runbook incident scenarios
 - compliance-sensitive workflows
-

9. Current Remaining Gaps

The baseline canonical package now exists under `docs/demo-readiness/`, but important follow-up artifacts and cleanups still remain.

Current remaining gaps:

- no canonical `00-program-map.md` yet
- no canonical `00a-deterministic-task-contract.md` yet
- no canonical `01-doc-inventory-gap-audit.md` yet
- `docs/API-SPEC.md` still needs explicit verification against real Ktor routes before it can be treated as canonical
- `docs/ARCHITECTURE.md` is still stale/conflicting and has not yet been rewritten or archived

Current hygiene issue:

- `infrastructure/README.md` contains secret-like example material and must be reviewed/cleaned

10. Working Rule for All Remaining PLOCK Work

Until a new canonical documentation package exists:

- use **repo reality first**
- use **BUILD-BLUEPRINT + README + RLS-AUDIT + runbooks** as architectural truth
- use **PRD + GTM + REGULATORY** for business, user, and market truth
- treat `docs/ARCHITECTURE.md` as stale
- treat agent narrative output as non-authoritative unless backed by real files

11. Decision

This document is the baseline authority map for all remaining PLOCK planning, QA, and implementation prep work.

Any future PLOCK task that produces:

- user stories
- feature lists
- ADR/BDR registers
- QA checklists
- implementation backlog

must explicitly reference this authority model before proceeding.

Revision #4

Created 2026-03-14 12:03:41 UTC by John

Updated 2026-07-12 20:01:49 UTC by John