Architecture at a Glance

Architecture at a Glance

How the resource is separated into editable configuration, public integration surfaces and protected inventory authority.

Applies to: `Nord Inventory v2.0.0`

Layers

Layer	Purpose	Typical files
Configuration	Owner-editable policy and defaults	`config/*.lua`
Shared definitions	Categories, items, weapons and common utilities	`shared/*.lua`
Bridges	Framework/notify/TextUI/target/lock compatibility	`client/bridge/*`, `server/bridge/*`
Runtime core	Inventory authority, persistence, sessions	`server/core/*`, `client/core/*`
Modules	Admin, commands, vehicles, public SDK	`server/modules/*`, `client/modules/*`
NUI	Player inventory and Admin Studio	`web/*`
Locales	PT/EN visible strings	`locales/*.json`
Database	Persistent registry/state/history	`sql/nord_inventory.sql`

Authority boundaries

- The **server** owns item state, slot state, money bridge operations, metadata, durability and storage sessions.
- The **client** owns presentation, input and a read-only local inventory snapshot used by client exports.

- The **NUI** never grants itself permissions; administrative mutations are revalidated by the server.
- Runtime providers register through explicit exports and are tracked by resource ownership.

Open vs protected release surface

The production package is designed for Tebex/Cfx Asset Escrow. Configuration, public bridge points, shared registries, locale files, SQL/reference files and item art stay owner-editable where appropriate. Core server authority remains protected.

Compatibility strategy

Integrations should prefer capability detection:

```
local caps = exports.nord_inventory:GetCapabilities()
if caps and caps.runtime_items then
    -- Register items dynamically.
end
```

This avoids assuming that every server uses the same optional features.

Revision #1
Created 2026-09-30 01:04:33 UTC by Riguita
Updated 2026-09-30 01:04:33 UTC by Riguita