MARINE TASK APP v2 — ULTRA COMPLETE ARCHITECTURE (MASTER)

PURPOSE

Single source of truth for architecture, phases, and implementation reality.

CORE PRINCIPLE

React renders. SQL scopes.

PERMANENT RULES

- Never attach tasks directly to yachts
- Never compute ownership in React
- Never overwrite task_results
- Never store tree paths
- Never use names as identity
- Never fetch entire datasets post-RPC

PHASE MODEL

Phase 1 — COMPLETE

UI + Templates:

React, Supabase Auth, TreeDisplay, Categories, Yachts, Groups, Templates.

Phase 2 — ASSIGNMENT (COMPLETE Feb 2026)

- Independent parent/child assignment
- Visual half tick
- Category move persistence
- Mobile-safe pencil
- task_category_map upsert
- Category editor with archive + rename
- Circular move prevention
- Virtual Top Level root

Phase 2.5 — FUTURE

Yacht Overrides:

tasks.yacht_id

task_categories.yacht_id

Merge or rename on conflicts.

Phase 3 — EXECUTION

task_context table anchors execution.

task_results immutable.

Phase 4 — RPC

get_task_tree()

get_yacht_tree()

Phase 4.5 — Indexing

Phase 5 — RLS

TASK LIFECYCLE

Template → Category → Yacht Assignment → Execution → Immutable History

TREE DISPLAY CONTRACT

UI only. No business logic.

CURRENT REALITY (FEB 2026)

Assignment UX:

Parent and child selectable independently.

Half-tick visual feedback restored.

No cascade.

Category Move:

Persisted via task_categories.parent_id.

Save button mobile safe.

Circular prevention active.

Mobile:

Pencil wrapped in DOM container.

Touch works reliably.

Persistence:

task_category_map uses UPSERT.

No duplicates.

Phase 2 core is complete.

NEXT PLANNED

GenericTreeAssignPage

GenericTreeEditorPage

Reusable for:

Categories

Yachts

User Groups

Same UX. Same logic. Different adapters.

FINAL SUMMARY

Template driven yacht maintenance system.

Phase 1 done.

Phase 2 done.

Phase 3+ future.

TreeDisplay remains pure.

SQL authoritative.