MARINE TASK APP v2 — ULTRA COMPLETE ARCHITECTURE (v3 — PHASE-GATED)

MARINE TASK APP v2 — ULTRA COMPLETE ARCHITECTURE (v3 — PHASE-GATED)

PURPOSE OF THIS REVISION

This document replaces ULTRA v2.

Goals:
• Remove ambiguity about when each system layer is introduced
• Explicitly mark temporary implementations
• Define Phase 2–5 execution order
• Avoid repeated architectural discussion in future chats
• Provide a single source of truth for ChatGPT + development

If something is not listed as "Current", it is considered future work.

CURRENT STATUS — PHASE 1 (COMPLETE)

Implemented:
• React + Vite shell
• Supabase Auth
• TreeDisplay.tsx (pure UI renderer)
• Task Templates
• Task Categories
• Yacht + Group Trees
• Virtual Unassigned Nodes
• Template Task Detail Pages

Current Technical Reality:
Hooks:
• useTaskTree
• useYachtGroupTree

Behavior:
• Load full datasets
• Join in React
• Resolve ancestry in JS

Temporary Phase■1 behavior only.

Rule:
React renders.
SQL scopes.

TREE DISPLAY

TreeDisplay.tsx remains UI■only.

Allowed:
• Selection
• Context menus
• Drag/drop
• Inline actions

• Icons
• Expand/collapse

Not allowed:
• Business logic
• Ownership resolution
• SQL knowledge
• Assignment logic

PHASE 2 — ASSIGNMENT

Introduces task_context.

```
create table task_context (
id uuid primary key default gen_random_uuid(),
task_id uuid references tasks(id),
category_id uuid references task_categories(id),
yacht_id uuid references yachts(id),
unit_override uuid,
period_override uuid,
created_at timestamptz default now(),
unique(task_id, category_id, yacht_id)
);
```

Adds assignment UI.
Context row created on assignment.

React joins still allowed.

PHASE 3 — EXECUTION

task_results table:

```
create table task_results (
id uuid primary key default gen_random_uuid(),
task_context_id uuid references task_context(id),
yacht_id uuid references yachts(id),
value numeric,
remarks text,
created_at timestamptz default now()
);
```

Results are immutable.

PHASE 4 — RPC + PERFORMANCE

Replace React joins with:
get_task_tree()
get_yacht_tree()

SQL becomes authoritative.

PHASE 4.5 — INDEXING

• group_links(parent_id)
• yacht_group_links(group_id)
• task_context(yacht_id)
• task_results(task_context_id)

PHASE 5 — RLS

Applied only after execution paths finalized.

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

TASK LIFECYCLE

1 Create template
2 Assign category
3 Assign yacht → task_context
4 Crew submits → task_results
5 History immutable

SUMMARY

Phase 1 UI/templates
Phase 2 assignment
Phase 3 execution
Phase 4 RPC
Phase 4.5 indexes
Phase 5 RLS

TreeDisplay remains pure.
SQL becomes authoritative.
Supports unlimited yachts and crews.