ARCHITECTURAL_HANDOFF.md

Date: 2025-12-12

Status: Phase 3 Complete / Entering Phase 4 (Analytics Refactor)

1. Context & State

The system has successfully transitioned from a CLI tool to a Web Application (Phase 3).

- **Current Capability:** Users can upload contest logs (Cabrillo). The system identifies the contest, applies complex scoring rules (via `score_calculators`), and displays a ”Score Summary” table in the browser.
- **Architecture:** Django Web App interacting with a ”Data Abstraction Layer” (DAL) in `contest_tools`. The DAL aggregates data; `views.py` is ”dumb.”

2. The Strategic Pivot (Phase 4)

We halted the plan to simply ”add fetching” and instead pivoted to a **”Strategy First”** analytics architecture.

- **Deprecation:** We deleted static plotting scripts (`plot_band_activity_heatmap.py`) because visualizations belong in the browser (dynamic), not as static PNGs.
- **Hierarchy:** We adopted a ”Top-Down” UI strategy:
  1. **The Arena (N > 2):** High-level fleet view. ”Who won and how?”
  2. **The Duel (N = 2):** Direct head-to-head comparison. ”Where did I lose?”
  3. **The Microscope (N = 1):** Self-analysis. ”Did I maintain rate?”

3. The ”Killer App”: Run vs. S&P Two-Way Breakdown

The most critical value-add identified is the **Two-Way QSO Breakdown**.

- **Concept:** Comparing two logs to see not just *what* they worked (Unique/Common) but *how* they worked it (Run vs. S&P).
- **The ”Common” Bar Logic:**
  - **Green (Run):** Both stations worked the target while Running (Shared Dominance).
  - **Blue (S&P):** Both stations worked the target while S&P (Shared Scraps).
  - **Mixed:** One Run, One S&P.

- **Strategic Insight:** Score separation at the top tier (Multi-2/Multi-Multi) comes almost exclusively from **Unique Run QSOs**. Efficient S&P is merely the ”price of admission.”

4. Immediate Roadmap (Next Actions)

We established a plan to ”Work Down” from the Dashboard.

Step 1: The ”Strategy Board” (Pending)

Upgrade the main Dashboard table (”The Arena”) to show high-level style metrics immediately.

- **Action:** Modify `TimeSeriesAggregator` (DAL) to calculate `run_qsos` and `run_percent`.
- **Display:** Add ”Run %” and ”Run Count” columns to the main HTML table.
- **Goal:** Allow users to see ”72% Run” vs ”40% Run” at a glance before drilling down.

Step 2: The Drill-Down

Create the dynamic version of the QSO Breakdown Chart.

- **Action:** Port the logic from `chart_qso_breakdown.py` into a Django view/Plotly.js component.
- **Requirement:** Must support the ”Common Bar” logic described above.

5. Known Technical Debts / Fixes

- **`cty.dat`:** The country files are out of date, impacting the accuracy of ”Unique Multiplier” reports. This needs a refresh mechanism.
- **Missing Files:** The architect (AI) lost track of `chart_qso_breakdown.py` in the context window. This file contains the reference logic for the breakdown charts and must be re-read upon resumption.

6. Prompt for Resumption

”I am resuming the session. We are at the start of Phase 4. The immediate task is to execute the 'Strategy Board' upgrade: modifying the DAL and Dashboard to display 'Run %' and 'Run QSOs' in the main table.”