

Architecture Roadmap: The Unified Engine Migration

Version: 1.9.10 **Date:** 2025-12-10 **Status:** Roadmap Paused (Phase 2 Complete)

1. The Strategic Vision

Goal: Create a single analysis engine that powers both a Command-Line Interface (CLI) and a future Web Interface (Django). **Core Constraint:** "Write Once, Render Everywhere." Logic must not be duplicated between CLI and Web.

2. Architectural Decision Records (ADRs)

ADR-001: Data Abstraction Layer (DAL)

- **Decision:** All business logic must be extracted into `data_aggregators/`.
- **Constraint:** Aggregators must return **Pure Python Primitives**.

ADR-002: Unified Visualization (Plotly + Jinja2)

- **Decision:** Replace Matplotlib with **Plotly**. Use **Jinja2** for HTML generation.

ADR-007: Shared Presentation Layer

- **Decision:** The Django Web App must NOT have its own template directory.
- **Mechanism:** It must point to the existing `contest_tools/templates`.
- **Reason:** Ensures CLI HTML reports and Web Views are bit-for-bit identical.

ADR-008: Title Layout Protocol

- **Decision:** Multi-subplot reports with 2-line titles must use `margin=dict(t=140)` to prevent overlap.
-

3. Master Transition Timeline

Phase 1: Data Decoupling (COMPLETE)

- **Status:** Complete. DAL is operational.

Phase 1.5: Stabilization (COMPLETE)

- **Goal:** Synchronize documentation and fix "Phantom Dependencies."
- **Status:** Verified.

Phase 2: Visualization Standardization (COMPLETE)

- **Goal:** Replace Matplotlib with Plotly/Jinja2.
- **Constraint:** Must be completed BEFORE Web Foundation to ensure charts work in browsers.
- **Tasks:**
 - ☒ Create `plotly_style_manager.py`.
 - ☒ Convert `chart_point_contribution.py`.
 - ☒ Convert `chart_point_contribution_single.py`.
 - ☒ Convert `chart_qso_breakdown.py`.
 - ☒ Convert `plot_cumulative_difference.py`.
 - ☒ Convert `plot_point_rate.py`.
 - ☒ Convert `plot_qso_rate.py`.
 - ☒ Convert `band_activity_heatmap`.
 - ☒ Convert `comparative_band_activity`.
 - ☒ Convert `comparative_band_activity_heatmap`.
 - ☒ Convert `comparative_run_sp_timeline`.

Phase 3: The Web Foundation (Simplified)

- **Goal:** Initialize Django *Synchronously*.
- **Constraint:** **NO Celery/Redis yet.** The initial deployment must be a simple monolithic app to reduce complexity.
- **Tasks:**
 - Initialize Django Project.
 - Mount `contest_tools` as an app.
 - Wire up Views to Aggregators.

Phase 4: Advanced Features

- **Goal:** Async Processing & Persistence.
- **Tasks:**
 - Introduce Celery/Redis (Only if processing time > 30s).
 - User Authentication.