

WBL - SOFTWARE ARCHITECTURE

1. OVERVIEW

This document describes the architecture of the World Billiard League (WBL) software suite. The system is designed as a distributed, LAN-based platform composed of multiple Windows Presentation Foundation (WPF) applications that coordinate through a shared PostgreSQL database. The database functions as the authoritative operational state store for all live match activities.

Each application operates as a peer with clearly defined state ownership. Applications have exclusive write access to the variables they own, while other applications consume those variables in a read-only manner and react to state changes. This design enforces loose coupling, eliminates direct inter-application dependencies, and ensures predictable system behavior.

2. EXTERNAL ACCESS

Native PostgreSQL logical replication is used to selectively mirror a subset of tables from the local operational database to a secure, read-only online replica. This strictly one-way replication supports real-time scoring, player and team statistics, fantasy systems, analytics, and other fan-facing applications, while ensuring the LAN-based operational database remains the sole source of truth for all match-control data.

3. APPLICATION ROLES

3.1 Broadcast Overlay System

The Broadcast Overlay System is a non-interactive, continuously state-driven application responsible for generating all on-air and in-venue broadcast graphics. In early deployments, this application also hosts the shared PostgreSQL database; this responsibility may be separated into a dedicated database host in later phases for fault isolation.

The application dynamically renders graphics using XAML, composites them onto the Program Output video feed, and outputs finalized SDI feeds for broadcast distribution and arena display. Supported graphics include dynamic score tickers, commentary-selected statistics, and analyst-driven overhead diagrams. A live difference-matte keying algorithm separates the player, balls, and table from the overhead camera feed and recomposites them with diagrams layered between those elements.

Key processes:

- ▶ Program Output video ingest via SDI
- ▶ Overhead camera ISO ingest via SDI
- ▶ SDI output of main broadcast graphics canvas
- ▶ SDI output of arena scoreboard graphics canvas
- ▶ Real-time graphical animation pipeline
- ▶ GPU-accelerated live keying using HLSL pixel shaders

3.2 Match Flow Controller

The Match Flow Controller is a user-operated control interface responsible for authoritative match progression. It manages match clocks, shot clocks, extensions, scoring, penalties, and team possession. This application is the sole writer for all timing, possession, and penalty-related state variables.

The application also interfaces with external systems, including radar hardware and venue lighting, to drive real-time environmental responses during match play.

Key processes:

- ▶ Overhead camera ISO ingest via SDI
- ▶ Radar hardware integration via COM interface
- ▶ Venue lighting control via DMX protocol

3.3 Commentary Statistics Interface

The Commentary Statistics Interface is operated by lead commentary personnel. It analyzes the current match context, including the active player, opponent, and shot conditions, to suggest relevant statistics such as shot probability metrics. This application does not compute or store authoritative statistics. Operators may explicitly select suggested statistics and designate them for on-air display via the Broadcast Overlay System.

Key processes:

- ▶ Match context analysis
- ▶ Statistical relevance evaluation
- ▶ Shot probability computation

3.4 Commentary Diagramming Interface

The Commentary Diagramming Interface is operated by a technical analyst and enables the real-time creation of animated visual annotations. These include trajectory lines, cue-ball contact indicators, and explanatory overlays rendered over the overhead camera feed. The application supports automated ball and pocket detection with snapping behavior and records clean table plates to support accurate live keying.

Key processes:

- ▶ Overhead camera ISO ingest via SDI
- ▶ Animated diagram rendering system
- ▶ Ball and pocket detection and snapping
- ▶ Difference-matte plate capture for clean table reference

3.5 History Editor

The History Editor is used to record and manage the outcome and table state of each shot, including precise event timing. It supports both live logging and retrospective editing of shot data. Edits performed within the History Editor trigger a full recomputation of player, team, season, and fantasy statistics. All retrospective edits are versioned and auditable.

Key processes:

- ▶ Overhead camera ISO ingest via SDI
- ▶ Ball and pocket detection and snapping
- ▶ Historical data editing and validation
- ▶ Full statistical recomputation pipeline
- ▶ Table geometry calibration for spatial analysis

3.6 Administrative Interface

The Administrative Interface provides database management tools for league operations and configuration. It supports management of match schedules, team rosters, player profiles, fantasy systems, and related configuration data. This application is used exclusively outside of live match operations and does not participate in real-time match control.

4. CURRENT DEVELOPMENT STATUS

At present, the core functionality of the WBL software suite is largely operational and validated through early development and testing. Most major features required for live match operations already exist in working form; however, several architectural and usability improvements are required to reach production readiness for a live broadcast environment.

The current implementation uses a MySQL database rather than PostgreSQL, and the entire feature set is consolidated into a single, monolithic application. This architecture must be refactored into the multi-application, shared-state design described in this document to support parallel operation, role-based control, and fault isolation during live events.

In addition, the existing user interface requires significant refinement to improve operator efficiency and clarity under live conditions. Several quality-of-life features, including robust ball detection and snapping, are incomplete or missing, and the shot probability model is currently only partially implemented and requires further development and validation.

5. SUMMARY

The WBL software suite is designed to support reliable, low-latency live match operations while enabling secure, scalable external data consumption. By combining single-writer state ownership, shared-database coordination, and selective data replication, the system balances operational integrity with flexibility and extensibility.