

Capital Forge League Updated Stadium Scoring + Interactive Stadium Developer Handoff

Executive Summary

This handoff consolidates the updated CFL scoring system reflected in the newly attached stadium assets and the Stadium Experience rules document. CFL is now explicitly a **multi-stadium business competition platform** driven by one universal Business Performance Points (BPP) engine, with stadium-specific movement and scoring visualization layered on top. [file:48] The code should therefore be refactored so that scoring logic is centralized, all stadium scoring values are stored in JSON, and the frontend reads that JSON to animate otherwise static stadium images into interactive game-style experiences.[file:48]

The most important implementation change is this: **do not hardcode football-first logic anymore**. The developer should treat football, basketball, baseball, golf, hockey, speedway, and soccer as stadium modules that consume the same verified business event feed but map those events into different movement units, special scoring triggers, penalties, and visual animations.[file:48]

This document gives Lavlesh the updated scoring tables, a normalized JSON configuration structure, Laravel/PHP service changes, event/broadcast guidance, and JavaScript instructions for turning the stadium images into motion-driven, console-style interactive views similar in feel to an Xbox or Nintendo sports game presentation.[file:48]

Source of Truth from the New Attachments

The attached Stadium Experience PDF establishes the official product architecture: one universal BPP engine, multiple selectable stadiums, round multipliers of Sweet 16 = 1.0x, Elite 8 = 1.25x, Final Four = 1.5x, and Championship = 2.0x, plus a rule that the stadium changes the presentation layer while the fairness model remains universal.[file:48] The attached scoring boards then add updated per-stadium scoring values and special trigger rules that must now be reflected in code.[file:48]

Key changes visible in the new stadium images include:

- Football now uses Investor reservation = 10 yds (First down) and Reserve Shares = Touchdown – 7 points with PAT / Extra Point = 1 point tied to deal room doc upload. [file:52]
- Basketball uses Investor reservation = 10 units (Paint touch), Reserve Shares = 3 Points (Slam Dunk), 50 Followers = 2 Points (Mid-range shot), and PAT / Deal Room Doc Upload = 1 Point (Free Throw).[file:50]
- Baseball uses Request for meeting = 1 base (RBI Single), Reserve Shares = Grand Slam – 4 Runs, 50 Followers = 2 Points (Stolen Base), and PAT / Deal Room Doc Upload = 1 Point (Bunt).[file:54]

- Golf uses Request for meeting = 1.0 (Birdie), Reserve Shares = HOLE-IN-ONE – Ace!, 50 Followers = 2 Points (Birdie Putt), and Deal Room Doc Upload = 1 Point (Tap-in). [file:53]
- Hockey uses Request for meeting = 10 units (Shot on goal), Completed presentation = 30 units cap (Goal – net bulge), and PAT / Deal Room Doc Upload = 1 Point (Wrist shot) plus a Penalty Box stall rule. [file:55]
- Speedway uses Request for meeting = 15 miles (Overtake), Completed presentation = 50 miles cap (Race-winning move), 50 Followers = 2 Points (Green Flag Pass), and PAT / Deal Room Doc Upload = 1 Point (Clean Lap) with a Rain Delay stall rule. [file:49]
- Soccer adds a new stadium not previously emphasized in older handoffs, with Completed presentation = 25 units cap (Breakaway goal), Reserve Shares = Hat Trick – 3 Goals, 50 Followers = 2 Points (Penalty Kick), and PAT / Deal Room Doc Upload = 1 Point (Corner Kick) plus a Red Card stall rule. [file:51]

Updated Universal Event Catalog

The backend should normalize all verified business actions into one universal event catalog before converting them to stadium movement or scoreboard points. [file:48] The recommended canonical event keys are:

```
{
  "project_page_view": "Unique verified project page view",
  "new_follower": "Authenticated user follows team",
  "daily_vote": "One verified vote per user per 24-hour window",
  "team_update": "Published team update",
  "veteran_pairing": "Approved veteran-athlete pairing milestone",
  "mentor_milestone": "Verified mentor/advisor milestone",
  "document_upload": "Verified business/compliance/pitch document upload",
  "ai_readiness_plus_10": "AI readiness score increases by 10 or more",
  "investor_reservation": "Reservation or equivalent investor intent event",
  "request_for_meeting": "Verified request to meet or equivalent investor engagement",
  "veteran_led_milestone": "Substantive milestone led by veteran partner",
  "milestone_badge": "Awarded badge for verified progress",
  "grant_or_prize": "Independent committee grant/prize award",
  "completed_presentation": "Completed investor/demo/presentation milestone",
  "reserve_shares": "Top conversion trigger mapped to stadium score event",
  "deal_room_doc_upload": "Doc upload that creates PAT / extra-point style scoreboard",
  "fifty_followers_bonus": "Triggered once at each 50-follower threshold"
}
```

This catalog should be stored in config and reused across all services so that a controller or observer fires canonical business events and the stadium adapter handles the sport translation. [file:48] [file:8]

Consolidated JSON Config Structure

Lavlesh should move all stadium scoring logic into JSON files, one per stadium, loaded by a single Laravel config file. The frontend should request this same config through an API so the visual layer always matches the backend rules.[file:48]

Master config layout

```
{
  "global": {
    "round_multipliers": {
      "Sweet 16": 1.0,
      "Elite 8": 1.25,
      "Final Four": 1.5,
      "Championship": 2.0
    },
    "stall_rules": {
      "football": { "name": "Stalled Drive", "effect": "movement_locked_until_new_" },
      "basketball": { "name": "Shot Clock Violation", "effect": "momentum_drop_until_" },
      "baseball": { "name": "Rain Delay", "effect": "game_suspended_until_new_mile_" },
      "golf": { "name": "Lost Ball", "effect": "stroke_penalty_until_new_milest_" },
      "hockey": { "name": "Penalty Box", "effect": "shorthanded_until_new_milesto_" },
      "speedway": { "name": "Rain Delay", "effect": "race_paused_until_new_milestor_" },
      "soccer": { "name": "Red Card", "effect": "reduced_strength_until_new_mile_" }
    }
  },
  "stadiums": {}
}
```

Football JSON

```
{
  "id": "football",
  "name": "Football Stadium",
  "movement_unit": "yards",
  "movement_cap": 25,
  "special_scoring": {
    "deal_room_doc_upload": { "points": 1, "label": "PAT / Extra Point" },
    "reserve_shares": { "points": 7, "label": "Touchdown" }
  },
  "events": {
    "project_page_view": { "movement": 1, "label": "Short gain" },
    "new_follower": { "movement": 1, "label": "Crowd push" },
    "daily_vote": { "movement": 2, "label": "Fan drive" },
    "team_update": { "movement": 3, "label": "Positive play" },
    "veteran_pairing": { "movement": 5, "label": "Formation set" },
    "mentor_milestone": { "movement": 5, "label": "Coaching play" },
    "document_upload": { "movement": 7, "label": "First-down setup" },
    "ai_readiness_plus_10": { "movement": 10, "label": "First down" },
    "investor_reservation": { "movement": 10, "label": "First down" }
  }
}
```

Basketball JSON

```
{
  "id": "basketball",
  "name": "Basketball Stadium",
  "movement_unit": "court_units",
  "movement_cap": 24,
  "special_scoring": {
    "deal_room_doc_upload": { "points": 1, "label": "Free Throw" },
    "fifty_followers_bonus": { "points": 2, "label": "Mid-range Shot" },
    "reserve_shares": { "points": 3, "label": "Slam Dunk" }
  },
  "events": {
    "project_page_view": { "movement": 1, "label": "Inbound pass" },
    "new_follower": { "movement": 1, "label": "Crowd energy" },
    "daily_vote": { "movement": 2, "label": "Fan drive" },
    "team_update": { "movement": 3, "label": "Positive possession" },
    "veteran_pairing": { "movement": 5, "label": "Set play called" },
    "mentor_milestone": { "movement": 5, "label": "Coaching timeout" },
    "document_upload": { "movement": 7, "label": "Half-court advance" },
    "ai_readiness_plus_10": { "movement": 10, "label": "Paint touch" },
    "investor_reservation": { "movement": 10, "label": "Paint touch" },
    "milestone_badge": { "movement": 15, "label": "Open look" },
    "grant_or_prize": { "movement": 15, "label": "And-one" },
    "completed_presentation": { "movement": 24, "label": "Fast break score" }
  }
}
```

Baseball JSON

```
{
  "id": "baseball",
  "name": "Baseball Stadium",
  "movement_unit": "bases",
  "movement_cap": 4,
  "special_scoring": {
    "deal_room_doc_upload": { "points": 1, "label": "Bunt" },
    "fifty_followers_bonus": { "points": 2, "label": "Stolen Base" },
    "reserve_shares": { "points": 4, "label": "Grand Slam" }
  },
  "events": {
    "project_page_view": { "movement": 1, "label": "Lead off" },
    "new_follower": { "movement": 1, "label": "Crowd energy" },
    "daily_vote": { "movement": 1, "label": "Single" },
    "team_update": { "movement": 1, "label": "Single" },
    "veteran_pairing": { "movement": 1, "label": "Walk" },
    "mentor_milestone": { "movement": 1, "label": "Walk" },
    "document_upload": { "movement": 2, "label": "Double" },
    "ai_readiness_plus_10": { "movement": 2, "label": "Double" },
    "request_for_meeting": { "movement": 1, "label": "RBI Single" },
    "veteran_led_milestone": { "movement": 2, "label": "Extra-base hit" },
    "milestone_badge": { "movement": 3, "label": "Triple" },
    "grant_or_prize": { "movement": 3, "label": "Triple" },
    "completed_presentation": { "movement": 4, "label": "Home Run" }
  }
}
```

```
}  
}
```

Golf JSON

```
{  
  "id": "golf",  
  "name": "Golf Stadium",  
  "movement_unit": "hole_progress",  
  "movement_cap": 2.0,  
  "special_scoring": {  
    "deal_room_doc_upload": { "points": 1, "label": "Tap-in" },  
    "fifty_followers_bonus": { "points": 2, "label": "Birdie Putt" },  
    "reserve_shares": { "points": 2, "label": "Hole-in-One" }  
  },  
  "events": {  
    "project_page_view": { "movement": 0.1, "label": "Chip shot" },  
    "new_follower": { "movement": 0.1, "label": "Chip shot" },  
    "daily_vote": { "movement": 0.2, "label": "Approach" },  
    "team_update": { "movement": 0.25, "label": "Fairway hit" },  
    "veteran_pairing": { "movement": 0.5, "label": "Birdie setup" },  
    "mentor_milestone": { "movement": 0.5, "label": "Birdie setup" },  
    "document_upload": { "movement": 0.75, "label": "Green in regulation" },  
    "ai_readiness_plus_10": { "movement": 1.0, "label": "Birdie" },  
    "request_for_meeting": { "movement": 1.0, "label": "Birdie" },  
    "veteran_led_milestone": { "movement": 1.0, "label": "Eagle setup" },  
    "milestone_badge": { "movement": 1.5, "label": "Eagle" },  
    "grant_or_prize": { "movement": 1.5, "label": "Eagle" },  
    "completed_presentation": { "movement": 2.0, "label": "Hole-in-One" }  
  }  
}
```

Speedway JSON

```
{  
  "id": "speedway",  
  "name": "Speedway Stadium",  
  "movement_unit": "miles",  
  "movement_cap": 50,  
  "track_length": 125,  
  "special_scoring": {  
    "deal_room_doc_upload": { "points": 1, "label": "Clean Lap" },  
    "fifty_followers_bonus": { "points": 2, "label": "Green Flag Pass" },  
    "reserve_shares": { "points": 1, "label": "Checkered Flag – Race Win" }  
  },  
  "events": {  
    "project_page_view": { "movement": 1, "label": "Pace lap" },  
    "new_follower": { "movement": 1, "label": "Crowd push" },  
    "daily_vote": { "movement": 2, "label": "Fan surge" },  
    "team_update": { "movement": 4, "label": "Segment gain" },  
    "veteran_pairing": { "movement": 8, "label": "Pit strategy" },  
    "mentor_milestone": { "movement": 8, "label": "Pit strategy" },  
    "document_upload": { "movement": 12, "label": "Lap gain" },  
  }  
}
```

```

    "ai_readiness_plus_10": { "movement": 15, "label": "Overtake" },
    "request_for_meeting": { "movement": 15, "label": "Overtake" },
    "veteran_led_milestone": { "movement": 10, "label": "Draft move" },
    "milestone_badge": { "movement": 25, "label": "Segment win" },
    "grant_or_prize": { "movement": 15, "label": "Pit road advantage" },
    "completed_presentation": { "movement": 50, "label": "Race-winning move" }
  }
}

```

Hockey JSON

```

{
  "id": "hockey",
  "name": "Hockey Stadium",
  "movement_unit": "zone_pressure",
  "movement_cap": 30,
  "special_scoring": {
    "deal_room_doc_upload": { "points": 1, "label": "Wrist Shot" }
  },
  "events": {
    "project_page_view": { "movement": 1, "label": "Neutral zone entry" },
    "new_follower": { "movement": 1, "label": "Crowd energy" },
    "daily_vote": { "movement": 2, "label": "Fan drive" },
    "team_update": { "movement": 3, "label": "Offensive zone entry" },
    "veteran_pairing": { "movement": 5, "label": "Power play setup" },
    "mentor_milestone": { "movement": 5, "label": "Power play setup" },
    "document_upload": { "movement": 7, "label": "Slot position" },
    "ai_readiness_plus_10": { "movement": 10, "label": "Shot on goal" },
    "request_for_meeting": { "movement": 10, "label": "Shot on goal" },
    "veteran_led_milestone": { "movement": 10, "label": "Power play goal" },
    "milestone_badge": { "movement": 15, "label": "Breakaway" },
    "grant_or_prize": { "movement": 15, "label": "Penalty shot" },
    "completed_presentation": { "movement": 30, "label": "Goal – net bulge" }
  }
}

```

Soccer JSON

```

{
  "id": "soccer",
  "name": "Soccer Stadium",
  "movement_unit": "pitch_units",
  "movement_cap": 25,
  "special_scoring": {
    "deal_room_doc_upload": { "points": 1, "label": "Corner Kick" },
    "fifty_followers_bonus": { "points": 2, "label": "Penalty Kick" },
    "reserve_shares": { "points": 3, "label": "Hat Trick" }
  },
  "events": {
    "project_page_view": { "movement": 1, "label": "Throw-in" },
    "new_follower": { "movement": 1, "label": "Crowd energy" },
    "daily_vote": { "movement": 2, "label": "Fan drive" },
    "team_update": { "movement": 3, "label": "Positive possession" },

```

```

    "veteran_pairing":      { "movement": 5, "label": "Set piece" },
    "mentor_milestone":    { "movement": 5, "label": "Tactical sub" },
    "document_upload":     { "movement": 7, "label": "Half-field advance" },
    "ai_readiness_plus_10": { "movement": 10, "label": "Penalty box entry" },
    "request_for_meeting": { "movement": 10, "label": "Penalty box entry" },
    "milestone_badge":     { "movement": 15, "label": "Open shot" },
    "grant_or_prize":      { "movement": 15, "label": "Free kick on goal" },
    "completed_presentation": { "movement": 25, "label": "Breakaway goal" }
  }
}

```

Laravel Backend Refactor Instructions

Lavlesh should consolidate scoring logic into a single service rather than spreading it across controllers or stadium views.[file:8][file:48] The preferred architecture is:

1. config/cfl_stadiums.php loads all stadium JSON files.
2. CFLScoringService accepts a canonical business event and team stadium type.
3. The service resolves movement, scoreboard points, round multipliers, and special triggers from config.
4. The service writes results into cfl_business_events and cfl_stadium_scores.
5. The service broadcasts a frontend event for animation.

Recommended tables

```

Schema::create('cfl_business_events', function (Blueprint $table) {
    $table->id();
    $table->foreignId('team_id')->constrained()->cascadeOnDelete();
    $table->string('stadium_type');
    $table->string('event_key');
    $table->decimal('base_movement', 8, 2)->default(0);
    $table->decimal('round_multiplier', 8, 2)->default(1.0);
    $table->decimal('final_movement', 8, 2)->default(0);
    $table->integer('score_points')->default(0);
    $table->string('movement_label')->nullable();
    $table->string('score_label')->nullable();
    $table->json('meta')->nullable();
    $table->timestamps();
});

Schema::create('cfl_stadium_scores', function (Blueprint $table) {
    $table->id();
    $table->foreignId('team_id')->constrained()->cascadeOnDelete();
    $table->string('stadium_type');
    $table->decimal('movement_total', 10, 2)->default(0);
    $table->integer('score_total')->default(0);
    $table->integer('follower_count')->default(0);
    $table->timestamp('last_activity_at')->nullable();
    $table->boolean('is_stalled')->default(false);
});

```

```

    $stable->timestamps();
});

```

Scoring service shape

```

class CFLScoringService
{
    public function applyEvent(Team $team, string $eventKey, array $meta = []): array
    {
        $stadium = config("cfl_stadiums.stadiums.{$team->stadium_type}");
        $round = $team->current_round ?? 'Sweet 16';
        $multiplier = config("cfl_stadiums.global.round_multipliers.$round", 1.0);

        $event = $stadium['events'][$eventKey] ?? null;
        $special = $stadium['special_scoring'][$eventKey] ?? null;

        $baseMovement = $event['movement'] ?? 0;
        $finalMovement = round($baseMovement * $multiplier, 2);
        $scorePoints = $special['points'] ?? 0;

        // Threshold-derived triggers
        if ($eventKey === 'new_follower') {
            $newFollowerCount = $team->stadiumScore->follower_count + 1;
            if ($newFollowerCount % 50 === 0 && isset($stadium['special_scoring']['fifty_followers_bonus'])) {
                $scorePoints += $stadium['special_scoring']['fifty_followers_bonus'];
            }
        }

        // Save event, update aggregate, broadcast result
        // Return payload for frontend animation.
    }
}

```

Trigger Rules That Must Be Implemented

The attached assets add several special-case triggers that should not be left as copy-only artwork. Lavlesh should implement them as first-class rules:

- `reserve_shares` triggers a scoreboard play, not just movement, in football, basketball, baseball, golf, speedway, and soccer.
- `deal_room_doc_upload` triggers a PAT/free-throw/bunt/tap-in/clean-lap/wrist-shot/corner-kick point event depending on stadium.
- `fifty_followers_bonus` triggers a bonus score event in basketball, baseball, golf, speedway, and soccer.
- `completed_presentation` is now a top-tier cap event in basketball, baseball, golf, speedway, hockey, and soccer.

- `request_for_meeting` is now an explicit scoring event in baseball, golf, speedway, hockey, and soccer.[file:49][file:51][file:53][file:54][file:55]

Stall / Penalty Logic

The artwork adds game-stall logic that should be made interactive rather than remaining decorative.[file:49][file:51][file:53][file:54][file:55] The backend should mark a team as stalled if there is no new verified accomplishment for a configurable time window, such as 7 days.

Recommended implementation:

```
{
  "stall_detection": {
    "inactive_days": 7,
    "auto_clear_on_event": true
  }
}
```

Frontend state by stadium:

- Football: dim offense, show “stalled drive” banner.
- Baseball: show rain delay overlay and pause runner animations.[file:54]
- Golf: show lost ball / stroke penalty overlay.[file:53]
- Hockey: show penalty box visual and reduced player intensity.[file:55]
- Speedway: show race pause / yellow flag behavior.[file:49]
- Soccer: show red card effect with reduced squad intensity.[file:51]
- Basketball: show shot-clock violation / cold streak effect even if not explicitly pictured, to stay consistent with the multi-stadium framework.[file:48]

Frontend Interactivity: Make Static Images Feel Like a Console Game

The stadium images should not remain flat JPG backgrounds. They should become animated stadium scenes with layered overlays, HUD elements, markers, score popups, and smooth transitions driven by live scoring events.[file:48] The goal is a presentation style similar to an Xbox or Nintendo sports title: high-contrast overlays, responsive score bugs, animated crowd energy, particle bursts, and event callouts.

Frontend stack recommendation

- Laravel Blade or Inertia page renders stadium shell.
- A JSON endpoint returns stadium config, team state, and recent events.
- JavaScript reads config and maps event keys to movement paths.
- Use GSAP or Anime.js for timeline animation.
- Use Canvas or absolutely positioned DOM/SVG overlays for moving avatars, pucks, balls, cars, runners, and possession indicators.

JavaScript event payload

```
{
  "team_id": 17,
  "team_name": "Apex Advisors",
  "stadium_type": "speedway",
  "event_key": "completed_presentation",
  "movement": 50,
  "score_points": 0,
  "movement_label": "Race-winning move",
  "score_label": null,
  "movement_total": 108,
  "score_total": 6,
  "is_stalled": false
}
```

Frontend render rules

```
function handleStadiumEvent(payload, config) {
  updateHud(payload);
  animateMovement(payload, config);
  animateScore(payload, config);
  animateCrowd(payload);
  animateSpecialEvent(payload, config);
}
```

Stadium-specific animation ideas

- Football: move the team marker downfield, flash first-down chains, animate touchdown end-zone burst.[file:52]
- Basketball: move possession arrow, animate half-court to paint path, trigger dunk burst for reserve shares.[file:50]
- Baseball: advance runners on bases, animate scoreboard inning lights, trigger home-run arc for completed presentation or reserve-shares grand slam.[file:54]
- Golf: move the ball from tee to fairway to green to hole, trigger cup-drop and flag pulse for reserve-shares ace.[file:53]
- Speedway: move the car along a curved SVG track path, trigger overtake smoke trails and checkered-flag finish burst.[file:49]
- Hockey: move puck through zones, animate shot-on-goal streak, trigger goal lamp and net ripple for completed presentation.[file:55]
- Soccer: move ball from midfield into the box, animate free kick / hat-trick / breakaway goal sequences.[file:51]

Interactive layers over the static image

Lavlesh should split each stadium screen into these layers:

1. **Background layer:** the attached static JPG.
2. **Path layer:** invisible SVG path(s) defining movement routes.
3. **Entity layer:** team icons, puck, ball, car, runner, golfer marker.
4. **HUD layer:** score, movement totals, event feed, round multiplier, crowd meter.
5. **FX layer:** glow, confetti, smoke, rain, penalty tint, scoreboard flashes.
6. **Modal layer:** enlarged rules tooltip when user taps stadium scoring legend.

This avoids redrawing the entire image and lets the product feel interactive while keeping current artwork intact.[file:49][file:50][file:51][file:52][file:53][file:54][file:55]

Suggested Files for Developer

```
config/  
  cfl_stadiums.php  
  stadiums/  
    football.json  
    basketball.json  
    baseball.json  
    golf.json  
    speedway.json  
    hockey.json  
    soccer.json  
app/Services/  
  CFLScoringService.php  
app/Events/  
  StadiumScoreUpdated.php  
app/Http/Controllers/  
  StadiumController.php  
resources/views/stadium/  
  show.blade.php  
public/js/  
  stadium-engine.js  
  stadium-animations.js  
public/css/  
  stadium.css
```

API Response Recommendation

```
{  
  "team": {  
    "id": 17,  
    "name": "Apex Advisors",  
    "stadium_type": "basketball",  
    "round": "Elite 8"  
  },  
  "scoreboard": {
```

```

    "movement_total": 41,
    "score_total": 5,
    "follower_count": 63,
    "is_stalled": false
  },
  "recent_events": [],
  "config": {}
}

```

The frontend should never hardcode event labels or point values; it should render them directly from the returned config and recent event payloads.[file:48]

Concrete Instructions for Lavlesh

1. Replace old hardcoded sport scoring with config-driven JSON loaded into Laravel.[file:48]
2. Add soccer as a first-class stadium option, not an afterthought.[file:51]
3. Add support for new event keys: `request_for_meeting`, `completed_presentation`, `reserve_shares`, `deal_room_doc_upload`, and `fifty_followers_bonus`. [file:49] [file:50] [file:51] [file:53] [file:54] [file:55]
4. Add aggregate score tracking separate from movement tracking, because several stadiums now produce both movement and scoreboard points. [file:49] [file:50] [file:51] [file:52] [file:53] [file:54] [file:55]
5. Implement threshold bonus logic for 50 followers. [file:49] [file:50] [file:51] [file:53] [file:54]
6. Implement stall detection and stadium-specific penalty overlays. [file:49] [file:51] [file:53] [file:54] [file:55]
7. Broadcast every score event to the frontend using Laravel events / Echo or polling if real-time is not yet available. [file:48]
8. Build a JavaScript stadium engine that reads JSON config and animates overlays on top of the existing JPGs. [file:48]
9. Keep all legal/compliance copy separated from gameplay logic; CFL remains educational and distinct from StateCF capital formation. [file:48]
10. Make the UI feel like a sports video game: animated HUD, motion, event banners, camera-like transitions, and satisfying feedback for every verified business accomplishment. [file:49] [file:50] [file:51] [file:52] [file:53] [file:54] [file:55]

Final Recommendation

The attached assets are now detailed enough that CFL should move from “illustrated concept” to a real config-driven stadium engine. The correct technical direction is one universal business event system, one JSON-based multi-stadium scoring framework, one Laravel scoring service, and one JavaScript stadium renderer that transforms the current static stadium art into interactive gameplay scenes. [file:48] [file:49] [file:50] [file:51] [file:52] [file:53] [file:54] [file:55]

This is the cleanest way for Lavlesh to implement the new scoring changes without future rewrites. Any future scoring update should require editing JSON and event mappings rather than rebuilding controllers or redesigning stadium views.[file:48]