ChatGPT

Building a Proprietary College Baseball Sabermetrics and Performance Analytics API

Executive summary

You want an API and codebase that other people can build on **without you inheriting the two biggest failure modes in sports data products**: (a) silently wrong numbers, and (b) "we don't actually have the rights to resell this." NCAA baseball is both **highly valuable** (scouting, media, coaching, betting-adjacent integrity) and **structurally messy** (fragmented stat collection, multiple scoring platforms, uneven tracking penetration, and strict platform policies). NCAA's own policies also center "official stats" around the host SID workflow, and they reserve the right to withhold publication until accuracy is substantiated—so your product has to treat "official" as a *process*, not a label. [1]

What this report concludes:

A credible proprietary college baseball analytics API is best built as a **three-layer system**:

1) **Rights + provenance layer** (what you're allowed to redistribute; where each field came from; what "official" means for this field and game). NCAA.com's Terms treat "statistics" and "updated scores" as NCAA Content in the Services and restrict commercial exploitation; NCAA policies define how official stats are produced and changed; and many premium sources contractually restrict redistribution. [2]
2) **Canonical data layer** (IDs, schema, event model, reconciliation, replay/backfill). NCAA single-game procedures and policies create specific invariants you can validate against (plate-appearance proving, host-control rule, "changes within one week," etc.). [3]
3) **Analytics layer** (context-adjusted metrics + probabilistic models), where you separate *descriptive* stats from *estimated* stats (e.g., xwOBA) and version every model. FanGraphs-style metrics provide stable reference definitions; MLB Statcast-style expected stats clarify required tracking inputs; academic baseball modeling shows how to shrink noisy samples and quantify uncertainty. [4]

Unknowns that materially affect architecture and business strategy:

- Target user types are unspecified (front-office/scouting, coaching staffs, media, fantasy, sportsbooks, fans, researchers all imply different latency, endpoints, and licensing).
- Budget is unspecified (data licensing cost can dominate infra cost; without a budget cap you need staged options).
- Desired licensing terms are unspecified (redistribution vs. "internal analytics only" drives pricing and compliance scope).

Data sources and licensing constraints

Official and semi-official statistical sources

NCAA's ecosystem matters because it defines what is "official," how corrections work, and what your ingestion latency realistically can be.

- **Official stats governance and correction workflow.** NCAA policies state that the statistics produced by the home team "should be considered the official account of the contest," and the visiting SID cannot alter statistics without consent of the home SID. They also outline that changes must come from school SIDs and should be made within one week for after-the-game press-box corrections, while NCAA staff may withhold posting until accuracy is substantiated. [1]
- **Submission mechanics (XML-centric).** NCAA policies explain that single-game stat files (XMLs) are used to create weekly rankings and populate results in RPI/NET and score reporting systems, and that XML files are uploaded via the stats.ncaa.org login "Game Reports" link and must follow weekly deadlines through the championship. [5]
- **Live stats display restrictions in championships.** NCAA's media coordination manual states NCAA.com is the official destination for live scores/statistics for championships and participating schools may not display live stats on their official websites except by linking to NCAA.com. [6]
- **Historical context.** NCAA hosts baseball statistics archives (e.g., compiled PDF rankings across divisions back to at least 1989–2000) and also provides a baseball records/resources page linking to record books, scorebook, and statistics policies. [7]

Practical data source categories you should treat as separate "rights domains":

- **Public NCAA-facing pages (leaders, archived rankings, record books):** good for context, but NCAA.com Terms and NCAA policies constrain reuse and commercial redistribution. [8]
- **School/conference scoring outputs (XML, box, PBP):** often what ultimately becomes "official," but you need either licenses/agreements with the originating entity or authorized access pathways. [9]
- **Third-party official distributors for specific use-cases:** e.g., NCAA has an extended partnership with Genius Sports [10] that positions NCAA LiveStats as the "official platform for live game statistics," notes it is free for member schools, and references scale (70,000+ games annually) as foundational infrastructure. [11]

Tracking, communications, and premium performance data

College baseball tracking is not universal; it's conference-, park-, and vendor-dependent, so your schema must support **partial coverage**.

- **Pitch/ball tracking adoption (conference examples).** Conference partnerships show TrackMan being used for "each pitch of every game" in conference play for umpire evaluation and related analytics workflows (ACC, Sun Belt, American). [12]
- ACC notes TrackMan uses Optically Enhanced Radar Tracking (OERT) and that conference teams can access scouting/analytics reports on league-issued iPads (with limits like no live strike zone access). [13]

- **On-field communications data.** PitchCom [14] announced a multi-year agreement integrating PitchCom data into Sportradar's Synergy Sports workflows starting with the 2026 college baseball/softball seasons, aimed at assessing pitcher command and better scouting context. [15]
- **Integrated coaching/scouting platforms.** 6-4-3 Charts positions itself as integrating sources "such as Synergy, TrackMan, and AWRE" into one platform and claims usage across hundreds of programs—evidence of competitive expectations for integration, not just raw stats. [16]

Third-party B2B data feeds for breadth and timeliness

If you want a redistributable API product, B2B feeds are usually the only scalable path—because their contracts are designed for downstream redistribution.

- Sportradar's marketplace describes a Global Baseball API product covering leagues "including … NCAA Baseball and Softball" with inning-by-inning updates, historical results, and team/player profiles. [17]
- Sportradar's developer docs emphasize their APIs are B2B and "not intended to be called directly from a client application," which strongly implies you should proxy via your backend and enforce your own auth/rate controls. [18]
- Sportradar also publishes "mapping feeds," including a player mappings endpoint that maps IDs between Global Baseball and the MLB API—useful as an existence proof for how professional providers solve cross-product ID reconciliation. [19]

Recruiting, roster, and biographical data sources

Recruiting data is often the most legally and contractually restrictive category. Many recruiting vendors monetize *the database itself* and explicitly restrict duplication and redistribution.

- Perfect Game USA [20] Terms of Use prohibit copying/reproducing, selling, and publishing content without written permission and limit use to personal, non-commercial contexts. [21]
- Prep Baseball Report [22] Terms of Service exist and generally follow paid subscription models; expect tight limitations around reuse (you must negotiate explicitly if you want redistribution). [23]

Core legal realities and constraints

In the U.S., raw sports facts are generally not protected as copyrighted expression, but *your risk is not only copyright*—it's contract terms, database rights (outside the U.S.), and "hot news"/misappropriation theories depending on how you source live data.

- U.S. Copyright Office states copyright does not protect "facts" (and also not ideas/systems/methods). [24]
- In *Feist v. Rural*, the Supreme Court held a phone directory's white pages were not entitled to copyright protection and reaffirmed that facts are not protected; originality is required. [25]
- NCAA.com Terms explicitly treat "statistics" and "updated scores" within its Services as "NCAA Content" that is owned/licensed and restrict commercial exploitation, reproduction, derivative works, and similar use without express permission. This is one of the clearest reasons that "scrape NCAA.com and resell" is not a viable foundation. [26]

- Web scraping legality is not the same as "permission to commercialize." The Ninth Circuit's *hiQ v. LinkedIn* analysis (CFAA scope for public pages) does not grant redistribution rights and contract claims can remain relevant. [27]

Bottom line for your product strategy: treat "official + redistributable" as something you buy/contract for, not something you scrape.

Architecture and data model

System design goals

The "why" of your architecture is to ensure you can simultaneously provide:

- **Truth:** reproducible, audit-able stats with provenance and correction history aligned to NCAA policy. [1]
- **Timeliness:** near-real-time where possible, but with explicit "provisional vs. official vs. corrected" states.
- **Coverage-aware analytics:** the system must gracefully handle missing play-by-play or missing tracking.

Recommended ingestion and ETL pipeline

This design assumes you will have (a) at least one licensed breadth feed (e.g., Sportradar product that includes NCAA baseball/softball), and (b) NCAA-governed "official" box/stat pathways for reconciliation/backfill. [28]

```
flowchart LR
  A[Licensed provider feed(s)\n(API / push / file drops)] --> B[Raw landing\nR2
object storage]
  A2[School/conference XML & box sources\n(authorized)] --> B
  A3[Tracking/comm data\n(TrackMan, PitchCom, etc.)] --> B

  B --> C[Staging parsers\nschema validation + normalization]
  C --> D[Canonical IDs & entity resolution]
  D --> E[OLTP store\nD1: entities, mappings, game headers]
  D --> F[OLAP store\ncolumnar warehouse: events + tracking]
  F --> G[Metrics engine\nbatch + incremental]
  G --> H[Metrics store\nmaterialized leaderboards,\nper-game/per-season
metrics]
  E --> I[Public API\nWorkers (Hono) + cache]
  H --> I
  F --> I

  I --> J[SDKs + dashboard\nDocs portal + usage analytics]
```

Key NCAA-driven operational constraints to reflect:

- **Host control & correction workflow:** treat the home team's report as official; away-team changes require consent; maintain a "corrections ledger." [1]
- **Game reconciliation invariants:** NCAA's proving box score rule implies you can validate whether a play-by-play/box is internally consistent (team plate appearances should equal runs + LOB + opponent putouts). [29]
- **Submission cadence and deadlines:** build "expected arrival windows" and late flags aligned to weekly deadlines and championship conclusion. [5]
- **When XML manual uploads occur vs. LiveStats scoring:** NCAA schedule/upload instructions note manual XML uploads are only allowed when the contest is marked "No" for NCAA LiveStats scoring or in sports not using LiveStats; this affects how you interpret missing XMLs. [30]

Canonical IDs and cross-source mapping strategy

This is one of the few areas where you can create durable proprietary advantage: everyone can compute wOBA, but not everyone can reliably unify "the same player" across NCAA/team/provider/tracking contexts.

Principles:

- **Persistent canonical IDs** for core entities: player, team, game, season. Canonical IDs never change; only mappings change.
- **Mappings are first-class data.** Keep a mapping table with: `source_system`, `source_entity_type`, `source_id`, `canonical_id`, `confidence`, `valid_from/to`, and a merge history.
- **Explicit merge semantics:** you will eventually need to merge two canonical players (e.g., data entry inconsistencies) and keep both mapping histories.
- **Use provider mapping feeds where available.** Sportradar explicitly provides "mapping feeds," including a Player Mappings endpoint for Global Baseball ↔ MLB ID mapping. [19]

Recommended data schema

The schema below is designed to support: (a) NCAA-style officially reported boxes, (b) play-by-play, (c) optional pitch/batted-ball tracking, and (d) historical archives.

Recommended schema (core tables and key fields)

Table	Purpose	Primary key	Key fields (non-exhaustive)	Notes
`source_system`	Identify each upstream source and its licensing domain	`source_system_id`	`name`, `type` (licensed/public/internal), `license_scope`, `allowed_redistribution`, `refresh_sla`, `contract_ref`	This is how you keep rights/provenance enforceable code.

Table	Purpose	Primary key	Key fields (non-exhaustive)	Notes
`canonical_team`	One row per program/ team identity	`team_id`	`ncaa_org_code` (nullable), `name`, `short_name`, `school_name`, `division`, `conference_id`, `home_venue_id`, `active_from/to`	NCAA schoo codes matte for loading files correctl [31]
`canonical_player`	One row per athlete identity	`player_id`	`full_name`, `bats`, `throws`, `primary_pos`, `dob` (nullable), `hometown` (nullable), `active_from/to`	Keep PII minimal; tre DOB as optional and access-controlled.
`canonical_game`	Game header	`game_id`	`season_id`, `date_start`, `team_id_home`, `team_id_away`, `venue_id`, `status` (scheduled/live/final/ suspended/forfeit), `doubleheader_n`	Suspended contests require spec handling for "official date site." [32]
`canonical_season`	Season context	`season_id`	`year`, `division`, `ruleset_version`, `start_date`, `end_date`	Used for run environment park factors, weights.
`entity_mapping`	Cross-source ID mapping	`mapping_id`	`source_system_id`, `entity_type`, `source_id`, `canonical_id`, `confidence`, `method` (direct/feed/heuristic), `valid_from/to`, `merge_group_id`	Mirrors provider "mapping feeds" concept. [19]
`roster_snapshot`	Rosters over time	`roster_id`	`team_id`, `season_id`, `as_of_date`, `player_id`, `class_year`, `jersey`, `is_active`	Rosters change; kee snapshots n overwrites.
`box_team_game`	Team box totals per game	`(game_id, team_id)`	`runs`, `hits`, `errors`, `lob`, `ab`, `bb`, `so`, `hbp`, `sb`, `cs`, `dp`, `tp`, `sf`, `sh`, `rbi`	Used for reconciliatio and "official" outputs.

Table	Purpose	Primary key	Key fields (non-exhaustive)	Notes
`box_player_batting_game`	Player batting line per game	`(game_id, player_id)`	`pa`, `ab`, `h`, `1b`, `2b`, `3b`, `hr`, `bb_uibb`, `ibb`, `hbp`, `so`, `sf`, `sh`, `r`, `rbi`, `sb`, `cs`	Align definitions t NCAA officia rules. [33]
`box_player_pitching_game`	Player pitching line per game	`(game_id, player_id)`	`ip_outs`, `bf`, `h`, `r`, `er`, `bb`, `ibb`, `hbp`, `so`, `hr`, `wp`, `bk`, `sv`, `hld`	Earned run reconstructi is defined by NCAA rules. [29]
`pbp_event`	Play-by-play event stream	`event_id`	`game_id`, `inning`, `half`, `outs_before`, `outs_after`, `bases_before`, `bases_after`, `batter_id`, `pitcher_id`, `event_type`, `result`, `runs_scored`, `rbi`, `is_error`, `fielders`, `timestamp`	Drives leverage, WP RE24, situational splits.
`plate_appearance`	Canonical PA (derived from PBP)	`pa_id`	`game_id`, `batter_id`, `pitcher_id`, `pa_result`, `bb_type`, `k_type`, `bip_type`, `woba_flag`, `li`	Stabilizes analytics eve if PBP source formats vary
`pitch`	Pitch-level data when available	`pitch_id`	`game_id`, `pa_id`, `pitcher_id`, `batter_id`, `pitch_type`, `velo`, `spin`, `loc_x/y`, `call`, `count_before/after`, `is_in_play`	Not always available; ke nullable and coverage-aware.
`batted_ball`	BIP-level tracking	`bip_id`	`game_id`, `pa_id`, `ev`, `la`, `spray_x/y`, `hang_time`, `hit_type`, `out_prob`	Enables xwOBA style expected stats. [34]
`winprob_state`	Win probability timeline	`wp_id`	`game_id`, `event_id`, `home_wp`, `run_expectancy`, `model_version`	Store model version to keep reproducibil

Table	Purpose	Primary key	Key fields (non-exhaustive)	Notes
`metrics_player_season`	Materialized season metrics	`(season_id, player_id)`	`woba`, `wrc_plus`, `ops_plus`, `babip`, `iso`, `k_pct`, `bb_pct`, `fip`, `xfip`, `k_minus_bb`, `war_est`, `quality_start_rate`, `model_versions`	Recompute nightly + backfill.
`park_factor`	Park/run environment adjustments	`(season_id, venue_id)`	`pf_runs`, `pf_hr`, `pf_1b`, `pf_2b`, `pf_3b`, `pf_so`, `sample_size`	College park vary massively; th is mandatory for “plus” metrics.
`linear_weights`	wOBA/run values per season/ division	`(season_id, scope)`	`wBB`, `wHBP`, `w1B`, `w2B`, `w3B`, `wHR`, `woba_scale`, `runs_per_pa`	FanGraphs weights change year you’ll derive your NCAA weights. [35]
`provenance_field`	Field-level lineage	`prov_id`	`table`, `field`, `source_system_id`, `method` (raw/derived/ model), `last_updated`	Lets you answer “whe did this number com from?” in the API.

Storage split:

- OLTP (D1): canonical entities, mappings, game headers, compact box.
- OLAP (warehouse): PBP/pitches/BIP and metrics computation.
  Cloudflare D1 has a per-database storage constraint (10 GB) and is designed for scale-out across multiple databases. [36]

Analytics and modeling layer

Advanced metrics to compute

Your metric set should be organized by **input requirements**—because college baseball coverage is not uniform.

Metrics computable from box score only (high coverage)

- **Rate stats:** AVG, OBP, SLG, OPS, ISO, BABIP

- BABIP = (H − HR) / (AB − K − HR + SF)
- **Plate discipline:**
- BB% = BB / PA
- K% = K / PA
- **Pitching components:**
- K/9, BB/9, HR/9, WHIP = (BB + H) / IP
- **Fielding-independent pitching:**
- FIP = ((13×HR) + (3×(BB + HBP − IBB)) − (2×K)) / IP + FIP_constant [37]
- You will estimate a *division/season* constant so that league-average FIP aligns to league-average ERA, following standard methodology. [37]

Metrics requiring play-by-play (medium coverage)

- **Run expectancy and RE24** (value above average given base/out state transitions).
- Requires every event's `bases_before/after`, `outs_before/after`, and runs scored. NCAA's "proving box score" rule provides a reconciliation invariant that helps validate this pipeline. [29]
- **Leverage Index (LI), WPA, clutch, win probability added**
- Requires a win probability model (state: inning/score/outs/bases + team strength if desired).
- **Situational splits** (performance by base-out, count, late/close, high leverage)
- **Baserunning value** (stolen base value, advancement on balls in play, etc.)

Metrics requiring tracking (low to uneven coverage)

- **Expected stats (Statcast-style):**
- xwOBA uses exit velocity and launch angle (and on certain batted balls, sprint speed). [34]
- You can train NCAA xwOBA/xBA models when you have enough TrackMan/HawkEye-like coverage; otherwise publish expected stats only for covered games and label the coverage.
- **Pitch modeling:** pitch quality, "stuff" grades, command metrics, release/shape features (where tracking allows).
- **Catcher framing (tracking-based):** needs pitch location + called strike model; many coaching/scouting platforms treat this as premium IP (expect licensing constraints). [38]

NCAA-specific context adjustments that matter

College baseball requires more aggressive contextualization than MLB:

- **Park effects** are larger and less stable (dimensions, altitude, scoring environments, humidors, ball variability).
- **Strength of schedule** variance is extreme (midweek scheduling, nonconference clusters).
- **Talent dispersion** affects model priors (freshmen volatility, transfer portal churn).

This is why "+" metrics (wRC+, ERA-, FIP-) should be computed on a division-and-season basis and reflect park + opponent adjustments. FanGraphs describes wRC+ as comparing wRC to league average while controlling for park effects, with 100 as league average. [39]

Modeling approaches for prediction and uncertainty

You should treat your analytics engine as two distinct products:

- **A measurement product** (metrics, leaderboards, splits) that must be reproducible.
- **A forecasting product** (projections, win probabilities, talent translation) that must be probabilistic and versioned.

Recommended model families:

- **Hierarchical Bayesian shrinkage** for rate stabilization (K%, BB%, HR rate, BABIP, etc.), especially early season and for players with limited PAs. A published hierarchical Bayesian model for MLB hitting demonstrates how to share information across time and players and incorporate covariates like age and position. [40]
- **OpenWAR-style transparent WAR framework** for a publishable "college WAR" that prioritizes reproducibility and interval estimates (uncertainty bands), rather than pretending WAR is a single exact number. [41]
- **Markov chain / state-based run models** for run expectancy and win probability (base/out/score states). Markov chain methods are widely used in baseball analytics and have formal treatments in academic work. [42]
- **Event outcome models** for batter/pitcher matchups, e.g., Bayesian hierarchical log5 approaches to predict matchup event probabilities with limited direct matchup data. [43]
- **Tracking-driven ML models** (gradient boosting / neural nets) for xwOBA-type estimators where EV/LA (and optional sprint speed) exist. [34]

Validation and accuracy checks

Your credibility will come from surfacing your validation rules openly.

Hard checks (must always pass):

- **Box score proofing constraints** per NCAA rules (team PA reconciliation). [29]
- **Home-team official control + change history** (do not overwrite; append corrections with who/when/why). NCAA requires official changes come from SIDs; away-team changes require home consent. [44]
- **Cross-table conservation:** runs in line score = sum of run-scoring events in PBP = team runs in box.

Soft checks (flag, don't block):

- Sudden roster anomalies (duplicate jersey, missing batter in lineup)
- Stat outliers (e.g., implausible inning totals) that require substantiation—consistent with NCAA's ability to withhold posting until accuracy is substantiated. [44]

Update cadence and backfill strategy

A realistic cadence aligned to NCAA workflows:

- **Live window (game time):** ingest "provisional" events/stats when you have a live feed (licensed provider or authorized LiveStats stream); publish with a `status: provisional`.
- **Post-game (T+0 to T+2h):** ingest final box; run proofing checks; compute preliminary metrics.
- **Correction window (T+1 day to T+7 days):** accept official corrections; rerun dependent metrics. NCAA policy points to "changes should be made within one week" for after-the-game corrections and emphasizes official change control. [44]
- **Historical backfills:** ingest archives (e.g., NCAA's PDF rankings sets) as snapshots and keep the raw PDFs in R2 for reproducibility and audit. [45]

API product and developer experience

API design goals

The "why" of the API contract is to make your platform usable to others **without leaking your internal complexity**:

- Clear separation between **raw facts** and **derived/model outputs**.
- First-class **provenance** and **coverage metadata**.
- Strong defaults: pagination, caching headers, idempotency, and stable IDs.

Authentication, rate limits, and tenancy

Given Sportradar's own guidance that B2B APIs are not intended for direct client calls, you should assume the same: your customers should call your backend, not upstream providers. [18]

Recommended approach:

- **Auth:** API keys for server-to-server; optional OAuth2/JWT for user-scoped products.
- **Tenancy:** `org_id` in auth context; enforce dataset entitlements (e.g., "tracking tier," "recruiting tier").
- **Rate limits:** token-bucket per API key with separate quotas for "heavy endpoints" (PBP, pitch tables).
- **Caching:** ETags + `Cache-Control` for immutable resources; short TTL for live endpoints.

Endpoint list with sample responses

API endpoints (v1)

Method	Endpoint	Purpose	Key query params	Sample JSON response (truncated)
GET	`/v1/health`	Service health	—	`{"status":"ok","uptime_s":123456}`

Method	Endpoint	Purpose	Key query params	Sample JSON response (truncated)
GET	`/v1/seasons`	List seasons/ divisions	`division`, `year`	`{"data":[{"season_id":"2026-d1","year": 2026,"division":"D1"}]}`
GET	`/v1/teams`	Team directory	`season_id`, `conference_id`, `q`	`{"data": [{"team_id":"t_123","name":"Texas","divisio`
GET	`/v1/teams/ {team_id}`	Team detail + context	`season_id`	`{"team_id":"t_123","name":"Texas","home_ve {"venue_id":"v_9"}}`
GET	`/v1/players`	Player directory	`season_id`, `team_id`, `q`, `pos`	`{"data":[{"player_id":"p_77","name":"First Last","team_id":"t_123"}]}`
GET	`/v1/players/ {player_id}`	Player bio + flags	`season_id`	`{"player_id":"p_77","bats":"R","throws":"R {"tracking_pct":0.42}}`
GET	`/v1/games`	Schedule/ results	`season_id`, `team_id`, `date_from`, `date_to`, `status`	`{"data": [{"game_id":"g_9001","date":"2026-03-01","s`
GET	`/v1/games/ {game_id}`	Game header	—	`{"game_id":"g_9001","teams": {"home":"t_123","away":"t_456"},"status":"f`
GET	`/v1/games/ {game_id}/ boxscore`	Official-ish box lines	`source`	`{"game_id":"g_9001","box":{"home":{"runs": 4}},"status":"official"}`
GET	`/v1/games/ {game_id}/ pbp`	Play-by-play events	`cursor`, `limit`	`{"game_id":"g_9001","next_cursor":"e_1200" [{"event_id":"e_1188","inning":7,"result":"`
GET	`/v1/games/ {game_id}/ winprob`	WP timeline	`model_version`	`{"game_id":"g_9001","model":"wp_v3","point [{"event_id":"e_1188","home_wp":0.74}]}`
GET	`/v1/metrics/ players`	Leaderboards by metric	`season_id`, `metric`, `min_pa`, `scope`	`{"metric":"wRC+","data":[{"player_id":"p_7`
GET	`/v1/metrics/ teams`	Team metrics	`season_id`, `metric`	`{"metric":"wOBA","data":[{"team_id":"t_123 0.391}]}`
GET	`/v1/players/ {player_id}/ splits`	Splits (PBP-based)	`season_id`, `split`	`{"split":"vs_RHP","pa":88,"wOBA":0.413}`

Method	Endpoint	Purpose	Key query params	Sample JSON response (truncated)
GET	`/v1/provenance/{resource}`	Explain lineage	`id`	`{"resource":"boxscore","sources":[{"source":"host_xml","updated_at":"2026-03`

Sample request/response patterns (expanded):

```
curl -H "Authorization: Bearer <API_KEY>" \
  "https://api.yourdomain.com/v1/games/g_9001/boxscore"
```

```
{
  "game_id": "g_9001",
  "status": "official",
  "provenance": {
    "policy": "host_official_stats",
    "last_corrected_at": "2026-03-04T18:22:10Z",
    "source_systems": ["host_xml", "licensed_feed_a"]
  },
  "teams": {
    "home": {"team_id": "t_123", "runs": 6, "hits": 9, "errors": 1, "lob": 7},
    "away": {"team_id": "t_456", "runs": 4, "hits": 8, "errors": 0, "lob": 6}
  }
}
```

The contract above encodes NCAA policy requirements—host official control and correction history—directly into the response. [1]

SDKs and client libraries

Recommendations:

- **TypeScript SDK** first (fits Workers/Next.js ecosystem), generated from OpenAPI.
- **Python SDK** next (analytics users), also generated.
- Version SDKs to API versions; support semver.
- Provide **typed pagination helpers** and **auto-retry with backoff** for 429/503.

Codebase structure and CI/CD

A Cloudflare-first monorepo pattern:

- `apps/api` — Workers API (Hono), routing, auth, rate limits, caching, OpenAPI.
- `apps/etl` — ingestion workers/queues consumers, parsers, validation, backfill jobs.
- `packages/schema` — canonical schema definitions + migration tooling.
- `packages/metrics` — metric formulas, linear weights derivation, model versioning.

- `packages/sdk-ts` and `packages/sdk-py` — generated clients + handwritten ergonomics layer.
- `apps/docs` — developer portal (static), OpenAPI reference, examples, changelog.

CI/CD essentials:

- **Schema migrations** gated by tests; investigate drift.
- **Replay tests**: re-run a fixed set of historical games and assert identical outputs (or explainable deltas).
- **Data contract tests**: "proving box score" checks for sample games. [29]
- **Observability**: per-endpoint latency, cache hit rate, error budgets.

Reproducibility:

- Version every metric and model (e.g., `woba_weights_2026_d1_v1`, `wp_v3`).
- Store raw source payloads in R2 with immutable keys and hash references.
- Preserve correction history, don't overwrite.

Privacy and security

Even "public sports data" can become sensitive when combined (tracking + health + academic timelines). Minimum controls:

- **Field-level access policies** (e.g., DOB, recruiting notes, internal tracking).
- **Audit logs** per API key: which resources were accessed, when, and at what volume.
- **Abuse controls**: rate limiting, bot detection, and anomaly alerts.

Also: NCAA's ecosystem is explicitly concerned with integrity and student-athlete safeguards in the context of official data distribution partnerships; you should align your product posture with that expectation. [11]

Roadmap and economics

Twelve-month roadmap from MVP to production

```
gantt
  title College baseball analytics API roadmap
  dateFormat  YYYY-MM-DD
  axisFormat  %b %Y

  section Rights and sourcing
  Vendor selection + licensing negotiations        :a1, 2026-03-01, 60d
  Data provenance policy + redistribution rules    :a2, 2026-03-15, 45d

  section Core data platform
  Canonical IDs + mapping framework                :b1, 2026-04-01, 70d
  Raw landing + parsers + validation gates         :b2, 2026-04-15, 90d
  Warehouse selection + OLAP schema implementation :b3, 2026-05-01, 75d
```

```
section Analytics engine
Box-derived metrics v1 + test baselines          :c1, 2026-06-01, 60d
PBP-derived metrics v1 (RE24/WPA/splits)         :c2, 2026-07-01, 75d
Context models (park + opponent adjustments)     :c3, 2026-08-01, 60d
Uncertainty & shrinkage models (Bayesian)        :c4, 2026-08-15, 75d

section Public API and developer product
API v1 design + auth + rate limiting             :d1, 2026-06-15, 75d
SDKs + docs portal + examples                    :d2, 2026-08-01, 60d
Private beta with 3-5 partners                   :d3, 2026-09-15, 75d

section Reliability and launch
Backfill pipelines + correction ledger           :e1, 2026-10-01, 60d
Load testing + cost tuning + caching strategy    :e2, 2026-10-15, 60d
Security review + key rotation + audit logging   :e3, 2026-11-01, 45d
Public launch + SLA tiering                      :e4, 2026-12-15, 60d
```

Cost estimate ranges and hosting options

Because budget is unspecified, the best way to stay honest is to give **parameterized ranges** and clearly separate **infra** from **data licensing** (often the largest and least transparent line item).

Cloudflare-first baseline (good for API + metadata + caching)

Core Cloudflare costs are predictable and published:

- Workers requests: $0.30 per million requests; CPU time: $0.02 per million CPU ms. [46]
- D1 (Workers Paid plan): first 25B rows read/month included + $0.001 per million rows; first 50M rows written/month included + $1.00 per million rows; storage billed per GB-month (rates depend on plan display). [47]
- D1 per-database storage limit: 10 GB; D1 is designed for scaling out across many smaller databases. [36]
- R2: published as zero egress fee object storage; pricing is based on storage volume and Class A/B operations, with an official calculator showing $0.015/GB-month and per-million operation rates (as of the calculator). [48]

Where this works: - Team/player directories, schedules, boxes, leaderboards, cached PBP slices.

Where this breaks: - Full pitch-level OLAP at scale (query patterns will be expensive and slow without columnar OLAP).

Hybrid: Cloudflare API + external OLAP warehouse (recommended for “premium advanced data”)

Two widely used OLAP options with published pricing references:

- **BigQuery on-demand:** first 1 TiB query processed per month free; then $6.25 per TiB processed. [49]

- **ClickHouse Cloud:** pricing page references $0.20/hr per compute unit and ClickPipes ingestion rates ($0.04/GB for ingested data). [50]

A realistic cost planning view:

- If your product serves mostly cached leaderboards and per-game boxes, Cloudflare costs can stay low.
- If your customers run heavy ad hoc queries (e.g., “all pitches for 3 seasons across conferences”), OLAP query cost dominates and you must enforce quotas or pre-aggregate.

Licensing cost guidance (cannot be responsibly priced here)

Most premium providers (and recruiting databases) price via custom quotes and contractual scope. This report can’t provide credible ranges without your target scope and vendor quotes; what it can do is tell you what *drives* the quote:

- redistribution rights vs “internal use only”
- historical depth (how many seasons)
- latency SLA (live vs T+1 day)
- granularity (box vs PBP vs pitch vs tracking)
- sublicensing rules, audits, and attribution requirements

Monetization and licensing models

Given the legal environment, your monetization should align to “what you own”:

- **Tiered API subscriptions** where the differentiator is (a) coverage, (b) latency, (c) advanced modeling outputs, and (d) redistribution rights.
- **Enterprise licensing** for redistribution or embedded products (apps/media).
- **Research tier**: lower volume, higher transparency, strict non-redistribution.
- **Data products**: periodic snapshots (season packs) with explicit license terms and hashes for reproducibility.

Competitive landscape and differentiation

The market expects integrated platforms combining stats + video + tracking + reporting:

- Conference-level TrackMan partnerships illustrate a direction toward pitch-by-pitch monitoring and in-game analytics access (ACC, Sun Belt, American). [12]
- Synergy/PitchCom integrations show how premium “context” data (communications, command evaluation) is being operationalized for college workflows. [15]
- 6-4-3 Charts markets integration across Synergy/TrackMan and emphasizes program adoption and specialized modules (e.g., framing). [16]

Your differentiation, if you execute correctly, is:

- **Trust layer:** correction ledger + provenance + reproducibility aligned to NCAA “official” process. [1]

- **ID layer:** best-in-class canonical mapping across NCAA/team/provider/tracking sources (explicit mapping tables, merge histories), similar in spirit to provider mapping feeds. [19]
- **Context layer:** park + opponent adjustments and uncertainty, not just point estimates. [51]

Legal and licensing checklist

Use this as a gate before you ship anything publicly.

Rights and contracts - Confirm whether each source is: public view-only, licensed for internal use, licensed for redistribution, or sublicensable. - For NCAA.com-derived content: do not redistribute without permission; NCAA.com Terms explicitly restrict reproduction/commercial exploitation of NCAA Content including statistics and updated scores. [26]
- For recruiting databases (Perfect Game, Prep Baseball Report): assume “no redistribution” unless contract explicitly grants it; Perfect Game terms restrict copying and commercial use without written permission. [52]
- For B2B feeds: ensure your contract explicitly covers downstream API redistribution, caching windows, and audit requirements.

Official-stat alignment - Implement host-official rule logic; maintain SID-originated corrections; enforce away-team-change constraints as metadata and workflow. [1]
- Encode the one-week correction expectation and preserve correction history. [44]

Compliance and governance - Provide provenance in API responses (source systems + timestamps + correction status).
- Protect potentially sensitive attributes (DOB, recruiting notes, tracking-derived medical/biometric inferences).
- Maintain an abuse and integrity posture consistent with broader NCAA data ecosystem expectations. [11]

Operational controls - Rate limits, key rotation, logging, and anomaly detection. - Document “provisional vs official” semantics and how backfills/corrections are handled.

Suggested visual diagrams to include in your developer portal

```
erDiagram
  CANONICAL_TEAM ||--o{ ROSTER_SNAPSHOT : has
  CANONICAL_PLAYER ||--o{ ROSTER_SNAPSHOT : appears_in
  CANONICAL_GAME ||--|| BOX_TEAM_GAME : has
  CANONICAL_GAME ||--o{ PBP_EVENT : contains
  PBP_EVENT ||--|| PLATE_APPEARANCE : groups_into
  PLATE_APPEARANCE ||--o{ PITCH : may_have
  PLATE_APPEARANCE ||--o{ BATTED_BALL : may_have
  SOURCE_SYSTEM ||--o{ ENTITY_MAPPING : provides
  ENTITY_MAPPING }o--|| CANONICAL_PLAYER : maps_to
  ENTITY_MAPPING }o--|| CANONICAL_TEAM : maps_to
```

```
flowchart TB
  S[Source payloads] --> V[Validators\n(proofing + invariants)]
  V --> M[Mappings\n(canonical IDs)]
  M --> C[Canonical store\n(entities + headers)]
  M --> W[Warehouse\n(events + tracking)]
  W --> A[Analytics engine\n(metrics + models)]
  A --> P[Public API\n(provenance + coverage)]
```

---

1 5 9 31 32 44 https://s3.amazonaws.com/fs.ncaa.org/Docs/stats/ForSIDs/Policies.pdf
https://s3.amazonaws.com/fs.ncaa.org/Docs/stats/ForSIDs/Policies.pdf

2 8 14 26 https://www.ncaa.com/tos
https://www.ncaa.com/tos

3 29 33 https://s3.amazonaws.com/fs.ncaa.org/Docs/stats/Stats_Manuals/Baseball.pdf
https://s3.amazonaws.com/fs.ncaa.org/Docs/stats/Stats_Manuals/Baseball.pdf

4 35 https://library.fangraphs.com/offense/woba/
https://library.fangraphs.com/offense/woba/

6 https://www.ncaa.com/_flysystem/public-s3/files/MC_Stats_Manual.pdf
https://www.ncaa.com/_flysystem/public-s3/files/MC_Stats_Manual.pdf

7 45 https://www.ncaa.org/sports/2013/11/21/baseball-statistics-archives-1989-2000.aspx
https://www.ncaa.org/sports/2013/11/21/baseball-statistics-archives-1989-2000.aspx

10 19 https://developer.sportradar.com/baseball/reference/global-baseball-player-mappings
https://developer.sportradar.com/baseball/reference/global-baseball-player-mappings

11 https://investors.geniussports.com/news/news-details/2025/NCAA-and-Genius-Sports-Expand-Partnership-Through-2032/default.aspx
https://investors.geniussports.com/news/news-details/2025/NCAA-and-Genius-Sports-Expand-Partnership-Through-2032/default.aspx

12 13 https://theacc.com/news/2025/3/7/baseball-acc-trackman-extend-partnership-through-2027.aspx
https://theacc.com/news/2025/3/7/baseball-acc-trackman-extend-partnership-through-2027.aspx

15 https://pitchcom.com/blogs/news/pitchcom-data-to-be-integrated-into-sportradar-s-synergy-sports-for-college-baseball-and-softball
https://pitchcom.com/blogs/news/pitchcom-data-to-be-integrated-into-sportradar-s-synergy-sports-for-college-baseball-and-softball

16 38 https://643charts.com/
https://643charts.com/

17 28 https://marketplace.sportradar.com/products/652fadc8459a02f489d9ff7c
https://marketplace.sportradar.com/products/652fadc8459a02f489d9ff7c

18 22 https://developer.sportradar.com/getting-started/docs/get-started
https://developer.sportradar.com/getting-started/docs/get-started

20 37 https://library.fangraphs.com/pitching/fip/
https://library.fangraphs.com/pitching/fip/

21 52 https://www.perfectgame.org/TermsOfUse.aspx
https://www.perfectgame.org/TermsOfUse.aspx

23 https://www.prepbaseballreport.com/legal/terms-of-service
https://www.prepbaseballreport.com/legal/terms-of-service

24 https://www.copyright.gov/help/faq/faq-protect.html
https://www.copyright.gov/help/faq/faq-protect.html

25 https://supreme.justia.com/cases/federal/us/499/340/
https://supreme.justia.com/cases/federal/us/499/340/

27 https://cdn.ca9.uscourts.gov/datastore/opinions/2022/04/18/17-16783.pdf
https://cdn.ca9.uscourts.gov/datastore/opinions/2022/04/18/17-16783.pdf

30 https://s3.amazonaws.com/fs.ncaa.org/Docs/stats/ForSIDs/Adding_Editing_Schedules.pdf
https://s3.amazonaws.com/fs.ncaa.org/Docs/stats/ForSIDs/Adding_Editing_Schedules.pdf

34 https://baseballsavant.mlb.com/statcast_field
https://baseballsavant.mlb.com/statcast_field

36 https://developers.cloudflare.com/d1/reference/faq/
https://developers.cloudflare.com/d1/reference/faq/

39 https://library.fangraphs.com/offense/wrc/
https://library.fangraphs.com/offense/wrc/

40 https://faculty.wharton.upenn.edu/wp-content/uploads/2009/11/Shanejensen.traj09.pdf
https://faculty.wharton.upenn.edu/wp-content/uploads/2009/11/Shanejensen.traj09.pdf

41 51 https://arxiv.org/abs/1312.7158
https://arxiv.org/abs/1312.7158

42 https://www.jstor.org/stable/171922
https://www.jstor.org/stable/171922

43 https://pmc.ncbi.nlm.nih.gov/articles/PMC6192592/
https://pmc.ncbi.nlm.nih.gov/articles/PMC6192592/

46 https://workers.cloudflare.com/pricing
https://workers.cloudflare.com/pricing

47 https://developers.cloudflare.com/workers/platform/pricing/
https://developers.cloudflare.com/workers/platform/pricing/

48 https://www.cloudflare.com/developer-platform/products/r2/
https://www.cloudflare.com/developer-platform/products/r2/

49 https://cloud.google.com/bigquery/pricing
https://cloud.google.com/bigquery/pricing

50 https://clickhouse.com/pricing
https://clickhouse.com/pricing