

HYDRA — Integration Brief

Audience: aggregator / operator integration teams · **Version:** 2.1 (game build v0.77) **Model:** supplier-hosted games, embedded on the operator's site, settled through the operator's wallet (seamless). **One adapter carries the whole catalogue** — HYDRA PULL and HYDRA MINORITY, each with solo and multiplayer modes, share one wallet and one integration surface.

1. HOSTING & EMBEDDING

HYDRA is hosted and operated by the supplier (Node.js, WebSocket transport with automatic reconnection). The operator embeds the games via an iframe or full-page redirect to a **launch URL** — either the games hall (both titles behind one door) or a specific game (`/pull` , `/minority`). Nothing is installed on the operator's side.

2. GAME LAUNCH FLOW

```
GET https://<game-host>/launch?token=<one-time-session-token>&currency=USD&lang=en&mode=real|demo&game=hall|pull|minority
```

1. The operator's platform authenticates the player and issues a one-time session token.
2. HYDRA validates the token against the operator's session endpoint and binds the connection to that player.
3. From then on the player is identified by session only — HYDRA holds **no registration, login, KYC, or funds**. Demo mode runs on supplier-side play credits and calls no wallet endpoints.

Reconnection and continuity: sessions survive refresh and network drops; open positions are never lost — in either game. PULL players abandoned mid-game are auto-cashed after a grace period (default 90 s), producing a normal credit transaction. MINORITY is asynchronous by design: a player may bet, close the app, and return later — the bet plays without them, the personal settlement is stored on the player's own record (later rounds never overwrite it), and open bets survive even a game-server restart, preserved together with the round's fairness commitment.

3. WALLET API (WHAT HYDRA CALLS)

HYDRA implements the aggregator's seamless-wallet specification during onboarding. Expected primitives:

Call	When	Notes
<code>balance</code>	on launch / on demand	display only; wallet remains authoritative
<code>debit</code> (bet)	the moment a bet is accepted	one transaction per bet
<code>credit</code> (win/cash-out/settle)	the moment money returns to balance	PULL: cash-out; MINORITY: round settlement, push and void refunds
<code>rollback</code>	on any unconfirmed/failed operation	idempotent by transaction id

All requests are signed (HMAC per aggregator spec), carry **idempotency keys**, amounts in minor units, and reference `roundId + transactionId`. Insufficient-funds and other error codes are surfaced to the player without side effects. One shared wallet covers both games — a player moves between titles with one balance and no internal transfers.

4. SETTLEMENT SEMANTICS (THE ONE THING TO KNOW)

Both games use **crash-category settlement**: money leaves the wallet at bet time and returns as a credit — no non-standard wallet features are required. The per-game shapes:

- **HYDRA PULL:** debit at bet, credit at cash-out. A position may span **multiple rounds** between those two events; its value evolves on-table and every intermediate change is recorded in the supplier's audit journal. Wallet lifecycle: exactly two operations (debit → credit).
- **HYDRA MINORITY (multiplayer):** debit per chip placed, credit at the round's settlement (win payout, push return, or void refund). Settlement may occur while the player is offline — the credit posts normally and the player's result board waits for their return. Cancels (pre-countdown only) emit refund-type credits.
- **Solo modes (PULL and MINORITY):** a single round-trip — debit at the play, credit with the result in the same interaction. No timers, no liquidity dependence; the simplest possible wallet shape.

Operator-triggered voids in either game emit refund-type credits at original stake value.

5. CURRENCIES, LIMITS, JURISDICTIONS

- Multi-currency per operator configuration; amounts and formatting follow the launch parameters.
- PULL table limits (max bet, max open exposure) are operator-configurable and displayed to players in real time. MINORITY runs flat-stake chip tables (default \$1–\$100 family); the chip set is operator-configurable.
- Market-specific responsible-gaming hooks (session limits, reality checks) are attached per jurisdiction at integration time.

— 6. FAIRNESS & COMPLIANCE PACKAGE

- All outcomes derive from CSPRNG-keyed HMAC-SHA-256. PULL multiplayer results are pre-committed on a 100,000-round hash chain; solo results are co-signed by a player-chosen client seed; MINORITY multiplayer commits to each round's tiebreak seed before betting opens (SHA-256 published at open, seed revealed at lock) — its win condition uses **no RNG at all**: the players' own bets decide, randomness only breaks exact ties. MINORITY solo commits a per-player seed before cells are marked and reveals it with each result. In-browser verification is built into every mode.
- The **Game Mathematics Document** (supplied, v2.1) contains the complete model for the whole family: zero-sum proofs, commission models, MINORITY's resolution and conservation theorems, solo RTP 95.2381% with a 10M-spin empirical check, bias bounds, and the disclosed-liquidity safety lemma.
- Every round of both games is written to an append-only, hash-chained journal (monthly archives, cross-file integrity), with per-round conservation residuals and phase-tagged drift attribution. Round and transaction CSVs export in one click for reconciliation and dispute handling.
- Liquidity policy, stated plainly: the public demo keeps MINORITY tables warm with clearly house-funded greeter bots (play-money, fully ledger-reported). **Real-money MINORITY uses no bots or simulated players**; liquidity is structural — asynchronous quorum rounds plus network-shared tables (players from every operator site sit at the same global tables).

— 7. BACK OFFICE (INCLUDED)

Real-time operator console for both games: commissions (rake / cash-out fee), table limits and chip sets, liquidity and computer-player controls, live rules editor, pause/maintenance, malfunction **void & refund** per game and per table, journal statistics and exports. Access can be scoped per operator.

— 8. ONBOARDING TIMELINE

Step	Owner	Typical duration
Sandbox launch URL + demo wallet stubs	supplier	days
Wallet adapter against aggregator spec (covers both titles)	supplier	2–4 weeks
Integration testing & sign-off	joint	1–2 weeks
Certification (if the target market requires)	lab	per lab schedule

Contact: Ilia Pochkhidze · ilia@hydrapull.com · Telegram [@iliapochkhidze](https://t.me/iliapochkhidze) · Live demo: hydrapull.com