

# Gruesøme's Arcade — Embedding Guide (v1.2)

Last updated: 2026-01-05

built by gruesøme

sig(enc,xor:0x5A,hex): 382f33362e7a38237a3d282f3f2999e2373f

## Purpose

This guide is for embedding any game inside the Arcade game library UI.

The Arcade is the parent app.

The Game runs in an iframe.

## 1) Embed contract (parent responsibilities)

Parent must:

- Create an iframe per game with a known gameld.
- Listen for ARCADE:READY from the iframe.
- Send ARCADE:SYNC snapshots periodically (or on relevant events):
- wallet connection changes
- credit balance changes
- membership changes
- nickname/avatar changes

Run lifecycle in embedded mode:

- Game sends ARCADE:REQUEST\_RUN
- Parent decides if it can grant:
- wallet connected?
- enough credits?
- PoH verified if required?
- Parent replies:
- ARCADE:RUN\_GRANTED { runId, costAC, eligible, rules }
- OR ARCADE:RUN\_DENIED { reason, message }

On game over:

- Game sends ARCADE:RUN\_RESULT { runId, score, durationMs, metrics, spentInRunAC? }

- Parent submits results to backend:
- `/api/run/submit` (authoritative ledger + leaderboards)

## 2) `postMessage` security

Recommended:

- channel key: a per-arcade constant (ex: "GA\_BRIDGE\_V1")
- origin:
- use referrer origin if possible
- otherwise accept "\*" but still require channel key

Parent should ignore:

- messages without the expected channel key
- messages from unexpected origin (when available)
- messages missing `runId` for `RUN_RESULT`

## 3) Catalog schema (`public/arcade-games.json`)

Each game entry should include:

- `id`: string
- `name`: string
- `embedUrl`: string (or relative path)
- `defaultMetric`: `metricId`
- `metrics`: [{

`id,`

`label?,`

`direction: "asc"|"desc",`

`clamp?: { min, max }`  
 `} ]`

- `usesCreditsInRun`: boolean
- `rankedSpendCapAC`: number (if `usesCreditsInRun`)

Metric IDs must exist in:

- `public/arcade-metrics-library.json`

## 4) UI integration expectations

Play view should show:

- run cost (paid/promo/free)
- whether the current run is payout-eligible (paid-only skill)
- metrics for the selected game (default metric + switchable metrics)

Leaderboard view should:

- allow selecting:
- game
- period (daily/weekly/all)
- board type (skill, activity)
- metric (from the game's metrics list)
- display identity:
- nickname (if set)
- PRO avatar hover card (if avatarPng exists)

Wallet view should:

- show daily + weekly payout countdowns
- show claimable breakdowns
- provide claim action buttons

## 5) Fail-safe UX rules

- If the game doesn't answer READY, show "Loading..." and allow reload.
- If RUN\_REQUEST times out (~2500ms):
- show "Connect wallet" or "Not enough credits — open Wallet"
- If the backend is unavailable:
- show local fallback leaderboard only (no payouts)

## 6) Admin / transparency note

If you embed admin/test accounts, mark them:

- excludedFromPayouts: true
- show on leaderboards (for transparency) but skip payouts and roll down rewards