PEER-TO-PEER ADS MARKETPLACE · SOLANA · SEPT 2026

Displaid.
Sell the pixels you own.
Get paid on proof.

A response to Josip Volarević's request for a startup. The marketplace half is easy; the half nobody had built is the escrow that only releases against verified display. That half is live on devnet.

displaid.pages.dev · Matthew Karsten · @expertvagabond

YOUR REQUEST, 7 SEPT

"Make a marketplace where anyone can sell a digital or physical location as ad space, and people can bid/buy it."

14.6K impressions, 45 replies, 46 bookmarks in a week. Then today: "If nobody builds this by mid next week, I'll be very tempted to just vibe code it myself."

You don't have to. Here is what exists, what it proves, and what ships before your deadline.

> Request for a startup: peer-to-peer Ads marketplace.
>
> — @solana just sold $167k of Ad space on its logo for Nepal flood relief
> — @FabianoSolana selling Ad space on his 𝕏 banner, $1.6k raised atm
> — TikTokers are selling their forehead as Ad space
>
> The behavior is already there ... Obviously crypto critical for global, cheap, and fast payouts.
>
> @JosipVolarevic2 · x.com/JosipVolarevic2/status/2096935262743900240

THE PART EVERYONE SKIPS

The marketplace is easy. Proof of delivery is the product.

How does a sponsor know the banner stayed up for the full seven days?

Cointraffic, AADS, Coinbound answer that by **being the middleman** — publisher-side, curated, 20–40% take. A peer-to-peer marketplace deletes exactly that role, and with it the only thing that made the money safe.

So every Solana-logo, Fabiano-banner, forehead deal today is struck on trust and settled by hand. It works between friends. It does not scale past them.

PAYMENTS

Commodity

USDC on Solana is the obvious rail. Every entrant will have it. It is not a moat.

LISTINGS + BIDS

A weekend

Any team in the Colosseum field can ship a grid of slots with a bid button.

PROOF OF DISPLAY

The moat

Escrow that **only releases against what a verifier actually observed**. Built, tested, running on devnet.

HOW IT WORKS

Four steps. The on-chain program only does custody and arithmetic.

01 · LIST

Seller lists a slot

A rectangle on their X banner or their avatar, priced per day. They prove control by putting `displaid:<code>` in their bio.

02 · BOOK

Buyer escrows USDC

Full amount into a vault PDA at booking. The booking stores a perceptual hash of the creative.

03 · VERIFY

Verifier watches the live profile

Every 5 min: fetch the real banner, crop to the slot, hash, sign pass / fail on-chain. One sample per interval, enforced by the program.

04 · SETTLE

Escrow splits on evidence

Anyone can crank `settle` after the window. Seller gets the delivered share, buyer gets the rest. No middleman, nothing stranded.

The program knows nothing about X, HTTP, or images. It trusts one key — the verifier — so upgrading the verifier's trust model is a config change, not a rewrite.

LIVE ON SOLANA DEVNET · VERIFIED END TO END

A real booking against a real banner, checked every five minutes.

7 SLOTS LISTED ON DEVNET	**156** /288 SAMPLES PASSED · 0 FAILED
0 bits HASH DISTANCE, LIVE BANNER	**100** TESTS GREEN · 11 RUST · 23 BANKRUN · 66 VERIFIER

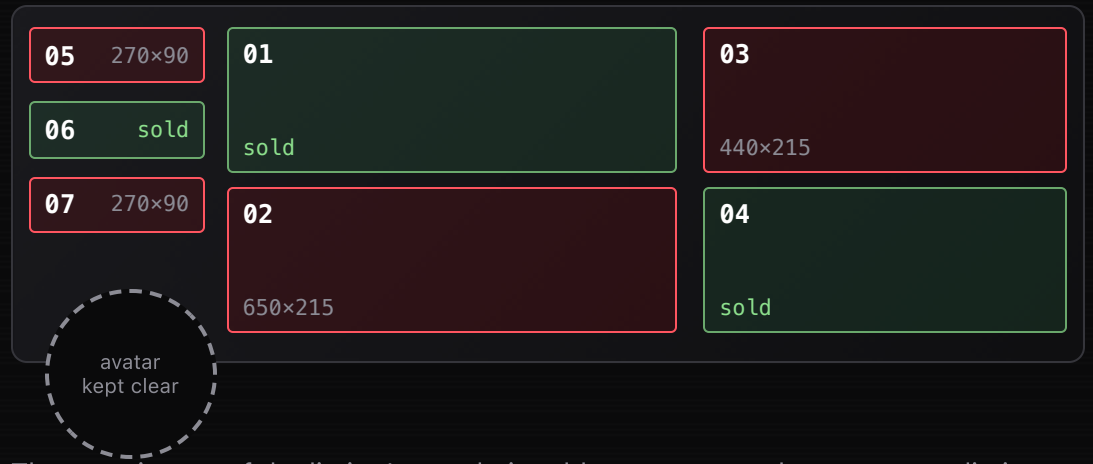
The booked creative is @expertvagabond's actual current banner (pHash `97606587df00794f`). The verifier fetched it from X, matched it at distance 0, and submitted signed samples that the page reads back from chain — no backend, no wallet, every figure decoded in the browser.

156 pass · 0 fail · 132 not observed (verifier process stopped 09-09 04:52Z) → settled on-chain 2026-09-14: seller 0.975 / fee 0.025 / refund 0 — exactly what the formula predicted

PROGRAM	4LU9iy7C…6x9XA
CONFIG	interval 300 s · grace 2 · fee 250 bps
BOOKING	8ngiZHCs…KTq5e · 1.000000 escrowed · settled
SETTLE TX	4Uax66aP…f5zRF
WINDOW	2026-09-08 13:15Z → 09-09 13:15Z
UI	displaid.pages.dev

ZONES · BUILT FROM YOUR MOCK-UP

One banner, seven sponsors, each verified independently.



The zone is part of the listing's on-chain address — seven slots are seven listings, seven escrows, seven independent checks.

Hashing the whole banner was wrong in **both** directions. Measured on a real 1500×500 banner:

CASE	WHOLE IMAGE	PER ZONE
own slot 05 (3.2%) removed	PASS · wrong	FAIL · right
neighbour resold, 05 intact	FAIL · wrong	PASS · right
own slot 01 (18.6%) swapped	FAIL	FAIL
X re-encode drift	0 bits	0–2 bits

Threshold calibrated on 7 live crypto-brand banners: re-encode drift ≤ 2 bits, separation between distinct creatives ≥ 20, median 30. The check has 4 bits of headroom either side.

SETTLEMENT POLICY

Three rules. Each one exists to stop a specific theft.

RULE 1

Denominator = samples observed

Verifier only managed 60 of 100 checks? Seller settles on 60. Charging a compliant seller for our downtime is not a fee, it is a bug.

RULE 2

Grace capped by delivery

`forgiven = min(failed, grace, passed)`. Drop the last term and a seller who showed nothing still collects the grace share. A grace budget becomes a theft budget.

RULE 3

Fee from seller's gross

The protocol never earns on money it is refunding. Zero observations → buyer refunded in full, protocol earns nothing, cost lands on whoever runs the verifier.

ABSTAIN

An outage is not a fail

Rate-limited or network dropped: nothing recorded. Account renamed or deleted: fail. Conflating the two was a griefing vector the live dry-run caught.

Conservation is structural: gross computed once, fee carved from it, refund is the remainder. Rounding dust lands in the refund and cannot escape.

WHO OWNS THE SURFACE?

Sellers prove control with the channel the verifier already reads: the profile itself.

- **Create listing** → it is born `verified: false` and cannot be booked.
- **Seller puts** `displaid:<first 16 chars of their seller key>` in their bio — one code covers every slot that key lists.
- **Verifier reads the bio** and attests on-chain. Only someone who can edit that profile can put the code there.
- **Revocable.** Handles get renamed, sold, abandoned. The code disappears → verification is withdrawn. Never on a transport failure.

WHY NOT HASH THE HANDLE?

Because that privacy was imaginary

We measured it: sha256 of every X handle that has ever existed is 0.1 core-hours on a laptop (2.26M hashes/s, pure Python). So handles are stored in clear and validated on write — `[a-z0-9_]{1,15}`, lowercase, no path characters — because the verifier interpolates them into an `api.x.com` URL.

Impersonation was never a theft vector (the creative is buyer-supplied, so samples would fail and the buyer is refunded) — but a buyer must not be induced to escrow against a surface the seller does not control.

THE FIRST USE

Raise your Breakpoint ticket from your own banner.

You said you have a way to help founders and creators raise money to attend Breakpoint. This is the app that makes those deals safe for the sponsor.

Creator lists seven slots for the four-week run-up. Sponsors escrow USDC. The creator is paid as each day is verified; the sponsor watches every sample land in public. Nobody chases anybody, and nobody has to already know and trust the creator to back them.

CREATOR

List 7 slots, keep the avatar

Bio code proves it is your handle. Price per day, minimum window. Done in five minutes.

SPONSOR

Escrow, then watch

USDC locked at booking. Pass / fail ticks accrue in public. Refund is automatic for anything not delivered.

ECOSYSTEM

A public record

Every booking, sample, and settlement is on-chain. Who sponsored whom, and whether it was delivered, is a fact — not a screenshot.

NEXT BUILD · YOUR "HIGHEST BID AT CLOSE" MODEL

Auctions are a small program change on top of the same escrow.

- **Cycle** = a listing's booking window. Slot opens for bids until close.
- **Bid** escrows the bid amount into a bid PDA. Outbid → refunded automatically, no action needed.
- **Close** is permissionless: the top bid becomes the booking, its escrow becomes the booking's escrow, every other bid PDA is closed with rent back to the bidder.
- **Everything downstream is unchanged** — sampling, settlement, the three rules. The auction only decides who books and for how much.

INSTRUCTION	STATUS
`initialize_config · create_listing · set_listing_active`	`live · zones on-chain since 09-14`
`verify_listing · book · submit_sample · settle`	`live`
`bid · close_auction`	`this week`
`wallet-connected book / bid UI`	`this week`
`mainnet + real USDC`	`next week`

Fixed-price listings stay — a creator who wants a known number should be able to name it.

WHAT IS NOT DONE · SAID PLAINLY

The verifier is a trusted oracle today. Here is the path off that.

TRUST

One key, one machine

Upgrade: run the *whole* fetch → crop → hash → sign loop inside a measured Intel TDX enclave (Attestbox / dstack). The verifier is pure TypeScript with no native binaries specifically so the container hashes identically on every build. Not claimed until it runs measured.

SURFACES

X banner + avatar

Pinned posts need a media lookup and can carry several images. Other networks and physical surfaces need their own fetcher. The program does not care — it only sees pass / fail / abstain.

SCALE

Full account scan per tick

Fine for one booking, throttled by public RPC before it is slow. Needs a memcmp filter on booking state plus pagination before real volume. Known, scoped, small.

EDGES

Overlapping zones · flat creatives

Nothing yet stops a seller listing two rects that intersect. Featureless creatives cannot be perceptually hashed and the UI should refuse them. Both measured, both documented.

COLOSSEUM · CRYPTO WORLD'S FAIR · 14 SEPT – 12 OCT

Registered. The field will ship marketplaces. This ships the settlement layer under all of them.

Judges will see a dozen slot grids with bid buttons. They will see one where the money provably cannot move without evidence, where the verifier's failure modes are priced into the payout formula, and where every claim on the pitch page is a devnet account they can read themselves.

The protocol is also usable by other marketplaces: any front end can list, book, and settle against the same program. "Winning" here means being the rail the category settles on.

WHAT IS PROVEN

Escrow → verify → settle, on devnet, against a real surface

100 tests. Live dry-run caught two bugs no fixture could, both fixed and covered.

WHAT SHIPS BY THE 20TH

Bids, wallet UI, mainnet USDC

Then the first ten creators list for Breakpoint.

PROPOSAL

Launch it for Breakpoint together.

You bring the demand side; I ship the rail.

- **This week (by 20 Sept):** bids + wallet booking UI on devnet. You review the flow as the first sponsor.
- **Next week:** mainnet with real USDC. First ten Superteam creators list their banner slots for the Breakpoint run-up — you pick them, I onboard them the same day.
- **Through 12 Oct:** every booking, sample, and settlement in public on the site. That record is the Colosseum submission.
- **Alliance:** once ten bookings have settled on-chain, the application writes itself. I'd like your intro then, not before.

WHAT I NEED FROM YOU

Three things, all this week

1 · Your reply on the RFS thread pointing at the live page.
2 · Names of the first creators who want to raise for Breakpoint.
3 · One sponsor willing to escrow the first real booking — $200 is enough to prove the loop end to end on mainnet.

Take rate is 2.5% of delivered value, from the seller's side only. Nothing is charged on refunds.

MATTHEW KARSTEN

Live page, source, and me.

Founder, Purple Squirrel Media. 4× blockchain hackathon winner, ETHDenver judge. Ships Rust / Anchor / TypeScript and the AI infrastructure around it — 78 MCP servers, 132 published packages.

▶ Live devnet page | @expertvagabond | karsten.cv

matthewkarstenconnects@gmail.com · program
4LU9iy7CrKgDpFkQBjzqfD9ycXSSu62As78JnZq6x9XA