METHOD PAPER

Decisions with receipts: the method (https://myoddly.com/method)

This is the citable methodology artifact for oddly: how observed commerce is turned into a number you can defend, and why every number on the public surfaces carries a receipt. It exists so that a reader who wants to cite oddly has one canonical source to point at, rather than re-deriving the method from scattered marketing copy.

It is written in two faces at once. A business reading leads with the outcome: what you can decide, and what risk is removed. A technical reading leads with the mechanism: how the number is computed and how the limit is enforced. A face is a temperature and an ordering, never a licence to relax the spine, so no sentence here drops a caveat the other face carries.

Nothing in this paper is a fresh assertion. Every figure and every mechanism resolves to an entry in the published claims ledger, and each section names the entry it stands on. If the ledger cannot receipt a number, the number does not appear here. The citation map at the end lists every claim in this paper against its receipt, so the acceptance test can be run by reading rather than by trust.

HOW TO CITE

Oddly Even Group. *Decisions with receipts: the method.* Published 2026-08-06. https://myoddly.com/method

Free to cite with attribution. A downloadable copy is published at decisions-with-receipts.pdf, and this page prints to the same paper.

1. The problem: decisions you cannot defend

THE OUTCOME

Most marketing decisions are made from information you cannot check, and the only thing that eventually tells you they were wrong is the profit and loss statement, months later, after the money is already spent. By then the budget is gone and the lesson is a guess about which of ten changes moved the number.

A decision you cannot defend is a decision you cannot repeat and cannot learn from.

THE MECHANISM

The failure mode is a broken feedback loop. An action is taken on a claim, the claim's provenance is not recorded, and the only signal that closes the loop is a lagging, aggregate, confounded financial outcome.

Attribution across that lag is underdetermined: many changes, one number, no isolation. The correction arrives too late and too coarse to update the rule that produced the decision.

The method in this paper is the answer to that problem: make every decision carry its receipt at the moment it is made, so the loop closes on the claim, not on the profit and loss statement. Nothing is claimed without a receipt.

2. The corpus: how observation works

THE OUTCOME

oddly's numbers come from real commerce it has actually looked at, and the counts on the site are the counts in the corpus today, growing week over week. There is no invented adoption number: until a figure is real and live, the stat is hidden rather than fabricated.

oddly re-checks a connected store every 15 minutes and still changes nothing without you.

THE MECHANISM

The corpus is served as aggregate counts only, with no per-store identity and no personal data, edge-cached, every count coalesced to zero with a retrieval stamp. A count that cannot be read reports zero and the stat is hidden, which is honest-null by construction.

The cadence is a scheduled trigger every 15 minutes that re-syncs the store and re-evaluates rules and playbooks. The loop proposes, it does not mutate. That cadence is stated as true because the trigger list says so at time of writing, not as a permanent property, and it is re-checked before use.

C4 The corpus counts are real and shown live: the public counts are read from the aggregate corpus surface, never hard-coded, and a read failure hides the stat rather than fabricating one.

Receipt: corpus number, fetched live. Aggregate only, no per-store identity.

C3 The 15 minute cadence is a scheduled trigger that re-syncs the store and re-evaluates merchant rules and playbooks.

Receipt: mechanism, verified before use. Re-read the trigger list before quoting the cadence.

Budget governance

THE OUTCOME

oddly can never quietly raise your ad spend, on any plan, and that limit has no upgrade. The downside you are underwriting when you connect an account is bounded to zero.

Nothing changes until you approve it. You see exactly what a change would do before it happens, and you can undo it inside the window.

THE MECHANISM

Ad-spend mutation is decrease-only by construction. A connector that could raise spend fails the build rather than shipping, so the asymmetry is a build gate, not a policy toggle a quarter could reverse.

The mutate path is propose, verify, dispose: dry-run preview, then human approval, then a reversible-within-window apply, each action traceable to the rule that fired it.

C1 Decrease-only ad spend is enforced as a build check: a spend-increase symbol in the connector or rules path fails the build.

Receipt: mechanism, enforced in code (brand-check rules 4 and 4a, scripts/brand-check.sh).

C2 Every mutating action runs a faithful dry-run preview, requires explicit human approval, and is reversible inside a known window.

Receipt: mechanism, published as measured counts on the proof page rather than asserted.

Longitudinal irreproducibility

THE OUTCOME

The corpus is small today and oddly says so. A handful of opted-in stores means many peer groups are still forming, and that absence is stated, not papered over.

The observation of a given week cannot be re-run later, because the commerce that produced it has already moved on. The honest record is the observation stamped with its date, not a claim of permanence.

THE MECHANISM

Observation is longitudinally irreproducible: a metric is the most recent observation carrying its own retrieval date, and a re-run observes a different world rather than reproducing the old one.

This is why the method fixes the receipt at observation time. You cannot go back and receipt a claim after the fact. The provenance is captured when the number is read, or it is lost.

3. The method: floors, honest nulls, provenance

Three mechanisms turn observation into a number you can defend.

k-anonymity floors

THE OUTCOME

oddly only tells you where you stand against a peer group when the group is real. If the cohort is still forming it says so, and no single store can be reverse-engineered from the comparison.

THE MECHANISM

A cohort number is served only under a floor of k greater than or equal to 5 distinct merchants paired with the spend bucket, and each contributor is reduced to an opaque hash before aggregation, so the comparison never touches a store's identity. Below the floor the cell is served as an honest null, never zero and never an estimate.

C6 The serving floor is k greater than or equal to 5 distinct merchants; contributors are hashed before aggregation; sub-floor cells return the honest-null dash.

Receipt: mechanism, docs/benchmark/cohort-key-decision.md. An unknown attribute stays in its own bucket and is never guessed.

Honest-null semantics

THE OUTCOME

When oddly has nothing to say, it says so, in one line, rather than filling the gap with a guess. "Cohort forming" is a real answer, not a softened estimate.

THE MECHANISM

Absence is a first-class value with its own rendering: a sub-floor cell returns the honest-null dash and gets a line on the index with no page of its own, and a corpus count that cannot be read reports zero and hides the stat.

A surface that goes quiet where it has nothing to say is making a claim, that it has nothing to say, and that claim is stated rather than implied.

Provenance chains

THE OUTCOME

Every move oddly names comes from a rule you can read, applied to a signal you connected, and the same inputs always produce the same output, so there is no black box to trust.

When oddly explains a move in plain words, the words can be model-written, but the decision and every number behind it are not.

THE MECHANISM

The rule catalog is the receipt: deterministic rules over connected sources, served from the registry so the page cannot drift from what the engine runs, with the decrease-only rule itself enforced as a build check.

The language layer is default-off and read-only over the verdict. It may rephrase two human-facing lines, never mutate a rank, a play, or a figure. The narration envelope enforces this by construction, because the writer never receives the claim.

C10 The published rule catalog is served from the same registry the engine runs, so the page cannot drift from the rules that fire.

Receipt: corpus number and mechanism, rendered live on how oddly thinks.

C11 The engine decides and the model only phrases: the generative layer sits behind a default-off flag and cannot add a play, move a rank, or touch a number.

Receipt: mechanism, the narration envelope. The writer is handed prose, never the claim.

Worked example A: the deterministic answer

THE OUTCOME

Ask oddly the same question five times and the answer does not move. If an answer moves, it was never a measurement, and a number that runs your ad budget should not move.

THE MECHANISM

On the published grounded eval, the oddly engine produced a byte-identical answer set across independent re-runs with zero fabricated claims, while language-model pipelines on the same closed-form store questions did not.

Carry the caveat always: the reproducibility is a structural property, because a calculator does not disagree with itself, and not independent brilliance. The interesting column is the models'.

C5 The engine is deterministic where language models are not, measured on a published eval and its reliability re-run.

Receipt: corpus number, docs/evals/. Stating the result as "we beat the frontier model" is a weaker and less honest claim, and is forbidden here.

Worked example B: a refusal, rendered honestly

The method is proven by what it refuses to say. A trust surface that records a security incident states that a breach was disclosed and declines to state a cause it did not observe, rather than inferring one to fill the field. The refusal is rendered as a first-class value, "we decline to state", the same shape as a sub-floor cohort's "cohort forming".

A method that can only produce answers is a method that will invent one. A method that can render a refusal honestly is one whose answers you can defend. The absence is the claim.

4. Receipts, defined

A receipt is evidence that would survive someone paid to disagree with you.

THE OUTCOME

oddly does not ask you to trust the pitch. Every strong claim points at the thing that backs it: a corpus number you can see, a mechanism you can read, or a stated absence.

If a claim has no receipt, it is removed, not softened.

THE MECHANISM

A receipt is one of exactly three forms: a corpus number with a path to its source, a named mechanism enforced in code (a build gate, a scheduled trigger, an envelope), or a named absence.

The adversarial test is the definition. Not "is this plausible" but "would this finding survive a reviewer whose job is to refute it". That is why the mechanism receipts are build gates rather than promises: a promise is refutable, a failing build check is not.

C8 Competitor claims are recorded as their claims, not verified fact, and are re-read against their live site on the day a comparison page ships.

Receipt: named absence and discipline. No unverified competitor claim ships on a public page.

5. The operator: built behind a human gate

The method's last receipt is the way the platform itself is built, which is the one part of the story a competitor cannot copy by rewriting their marketing.

THE OUTCOME

oddly is largely built by its own operator, running continuously, and every change it makes to the product is held behind a human gate before it ships.

The person approving is approving specific content, not signing a blank cheque. The accountability layer oddly sells is the same one it runs on itself.

THE MECHANISM

The operator is an autonomous build runner. It proposes changes as pull requests and cannot merge a gated class of change without a review approval, from an allowlisted approver, on the pull request's current head commit.

Consent attaches to content, never to a pull-request number: an approval on an older commit does not count. The gated classes are mechanical, not a judgement call, and they include a change to the operator's own instructions, so it cannot grade and release its own control plane.

C7 oddly runs its own stores on its own product, published as plays executed and whether the value covered the subscription.

Receipt: mechanism and measured counts on the proof page. Counts and multiples only, never a store's revenue or spend.

Gate The merge gate classifies every change and holds a gated class until a named human approves that exact commit.

Receipt: mechanism, enforced in code (automation/phase2/human-gates.ts and the gate-class check that runs on every pull request).

This is the same propose, verify, dispose shape as the product's mutate path, one level up. The agents propose and hold. A person disposes. The live map of that estate, with every node wired to a receipt you can open, is published at the estate map.

Citation map: every claim to its receipt

Read this section to run the acceptance test. Each row is a claim made above and the ledger entry that receipts it. If a future edit adds a sentence whose claim is not in this table, that sentence is the drift this method exists to prevent, and it must be removed or receipted before it ships.

SECTION	CLAIM IN THIS PAPER	RECEIPT
2 Corpus	Counts are real, live, aggregate-only, honest-null	C4
2 Corpus	Re-checks every 15 minutes, proposes not mutates	C3
2 Budget	Decrease-only ad spend, enforced as a build gate	C1
2 Budget	Nothing changes until you approve; reversible window	C2
2 Longitudinal	Corpus is small today and says so; cohorts forming	C6, named absence
3 Floors	k greater than or equal to 5 cohort floor, hashed contributors	C6
3 Honest null	Sub-floor cell and unreadable count render absence	C4, C6
3 Provenance	Deterministic rule catalog, decrease-only build check	C10
3 Provenance	Model phrases, never decides; narration envelope	C11
3 Example A	Byte-identical answers, zero fabrication, with caveat	C5

SECTION	CLAIM IN THIS PAPER	RECEIPT
3 Example B	Refusal rendered honestly (breach cause declined)	Named absence
4 Receipts	Claim with no receipt is removed, not softened	Ledger read-rule
4 Receipts	Competitor claims re-verified on ship day	C8
5 Operator	The proof page runs oddly's own stores on the product	C7
5 Operator	Human gate pinned to a commit; consent attaches to content	Merge gate

What this paper refuses to claim

Stating what we refuse to claim is itself a claim, and these are load-bearing.

- **No fabricated adoption numbers.** No store-count boast, no total merchandise volume, no placeholder star rating. Until a number is real and live, the stat is hidden, not invented.
- **We do not claim to beat the frontier model.** The eval result is a determinism claim carrying its own caveat, not a capability boast.
- **We do not serve a cohort below the k-anonymity floor.** "Cohort forming" is the honest answer, not a softened estimate.
- **We do not state a breach cause we did not observe.** Where the evidence is thin the surface says "we decline to state" rather than filling the gap.
- **The corpus is small today and we say so.** A handful of opted-in stores means many cohorts are still forming. That is stated, never papered over.

Where this paper comes from

This page renders a source paper that is authored and versioned in the same repository that builds the product, so the citation target is diffable and reviewable before any pixel is cut. The source lands first and the surface renders it, which is the same sequencing the claims ledger follows.

Every claim above resolves to that published ledger. A claim that the ledger cannot receipt does not appear on this page, in either face.

Published 2026-08-06. Related reading: how oddly thinks, the proof page, and the estate map.