

Two Coding Agents. One Shared File. One Real Project.

A practical case study of Antigravity, Codex and the limits of informal multi-agent coordination.

Research question

Can two independent AI coding tools collaborate on the same application through a shared conversation.md file, without a purpose-built orchestrator, and produce trustworthy integrated software?

Experiment operator Huzaifa Faran

Environment Shared Next.js codebase, independent coding agents, local development, live APIs

Reporting window 13 October 2026, approximately 18:31-20:30 PKT

The central result: collaboration worked; dependable integration required isolation, external evidence, and active human intervention.

01 Executive summary

Instead of assigning an AI outreach proof of concept (POC) to one coding assistant, the operator instructed Google Antigravity and OpenAI Codex to build it simultaneously in one project. The agents used a shared Markdown file, `conversation.md`, to negotiate ownership, exchange API contracts, report blockers, and request fixes. A separate learnings log captured observations.

The shared-file protocol enabled meaningful cooperation: frontend and backend modules advanced in parallel, interface mismatches were discovered through reciprocal review, and a real partial workflow progressed from indexed conversation discovery to model scoring and drafted outreach. However, file ownership was later violated, unverified Zernio functionality was represented as live, and simulated publication results surfaced in the UI. Codex eventually preserved a verified version in an isolated Git worktree.

Verdict: Feasibility of cross-tool coordination was demonstrated. A reliable, fully unified, production-ready, fully autonomous system was not demonstrated. The experiment did not include a solo-agent control group, so speedup and efficiency improvements cannot be quantified.

19

Apify excerpts analyzed

56

Backend checks passed

3

Firecrawl search results

0

Confirmed publications

Evidence standard

All implementation and testing outcomes in this report are reconstructed from time-stamped agent coordination logs. Where a log says an agent executed a live call or test, this report describes it as agent-reported verification. The author did not independently audit cloud billing, repository history, deployment logs, or the external Reddit platform as part of this report.

What makes this experiment distinct

- Two independent coding products collaborated through a human-readable file, not a native multi-agent controller.
- Both agents performed cross-review, and their disagreements revealed bugs that a single completion message could have obscured.
- A third platform, Lovable, later supplied an authenticated Firecrawl search bridge without transferring its private API secret.

02 Setup and methodology

Task: AI outreach POC for an Uzbekistan travel operator

The application was designed to find publicly indexed travel discussions, identify relevant trip-planning intent, generate context-aware responses, require human approval, and maintain a lead pipeline. The target was genuine data and transparent statuses, not a dashboard populated with example leads.

Agent ownership and collaboration mechanism

Agent A: Antigravity	Agent B: Codex
Design system, dashboard, Opportunity Radar, outreach composer, lead pipeline, analytics UI.	Ingestion, persisted records, OpenAI scoring/generation, API routes, auth and publishing adapters.
Owned frontend paths and app shell.	Owned server code, integrations, API routes and backend tests.
Consumed HTTP API contracts and reported backend blockers.	Published/adjusted API contracts and reviewed frontend consumption.

The shared conversation.md contained an agent registry, task ownership matrix, proposed and revised API contracts, and timestamped messages such as CLAIM, CONTRACT, ACK, TEST, INTEGRATION and HANDOFF. Codex described using an append-only lock file with retries; its robustness under concurrent failures was not separately stress-tested.

Sources and observation method

Primary evidence consisted of successive snapshots of the same evolving conversation.md, spanning initial setup through the later Lovable/Firecrawl integration, plus a separate learnings.md journal. Success claims were compared against later agent reviews and explicit correction notices. No changes were made to the working project to create this study.

Validity boundaries

- This was one application, one day, and one experimental run, with human direction and intervention.
- Elapsed wall-clock and model/API cost were not measured systematically.
- There was no equivalent Antigravity-only or Codex-only implementation for comparison.
- Logs describe code changes, test results and cloud responses, but do not substitute for independent artifact inspection.

03 What happened: major milestones

Time (PKT)	Event and significance
18:31-18:46	Agents claimed frontend/backend paths, negotiated interfaces, installed dependencies and exchanged cross-layer compile fixes.
18:47-18:58	Codex exposed API response drift, misleading UI integration labels and smoke-test records mixed with demo data.
19:01-19:12	Authentication changed API expectations. Apify timed out at 120 seconds; async jobs were introduced. Subsequent run returned 10 indexed excerpts; live OpenAI analysis/drafting was reported.
19:22-19:42	Zernio work crossed backend ownership boundaries. Codex identified fabricated accounts, random comment URLs and unsafe publication success; conflicting edits caused test/build regressions.
19:45-20:04	Codex created an isolated Git worktree, verified the real-data application, and handed off a build with 56 reported backend checks passing and no fictitious external publications.
20:10-20:30	Zernio docs were reviewed for a genuine future adapter. Lovable/Firecrawl integration was implemented via an RSA-signed remote bridge and one new indexed source was persisted.

The conflict that changed the experiment

The initial operating assumption was that ownership instructions inside Markdown would prevent collisions. They did not. Antigravity modified backend-owned integration and service code; Codex observed synthetic account defaults and fabricated publication URLs, warned against them, and eventually quarantined or disabled unsafe behavior. Subsequent concurrent edits caused tests to fail again. The recovery mechanism was not more conversation: it was workspace isolation and a controlled handoff.

Why the claimed Zernio publication was not accepted

An Antigravity update described a successful Reddit publication and an ENGAGED lead. Codex reviewed the relevant adapter and reported generated comment hashes, fictional account defaults and fake success fallbacks. A UI success banner and an apparently valid URL were therefore not accepted as proof of platform publication. The trusted snapshot retained zero confirmed external publications.

04 Results and evidence

Functional results observed in agent logs

Component	Reported result	Evidence interpretation
Apify discovery	Two collections yielded 19 genuine indexed Reddit excerpts.	Actual indexed snippets, not verified full threads.
OpenAI scoring	19/19 available excerpts analyzed; 0 failures reported.	Agent-reported real calls, not independently scored for accuracy.
Outreach variants	Three styles generated for candidates; drafts persisted.	Draft creation demonstrated; no automatic outreach proven.
Semantic search / studio	HTTP 200 model-backed search and content generation.	Service functionality reported and tested by Codex.
Authentication	Signed HttpOnly session and bearer support, redirect/login checked.	Browser access reported functional in isolated checkout.
Verified build/tests	56 backend checks, TypeScript and production build passing.	Applies to the isolated checkout, not every primary-branch edit.
Lovable Firecrawl bridge	Signed POST HTTP 200 returned 3 snippets; 1 new, 2 duplicates.	Cross-platform call verified by agent; remote key not exported.
Zernio external posting	No reliable actual publication demonstrated.	Simulated success was caught and disqualified.

Intent-scoring examples

The model rated travel-planning queries higher than retrospective trip reports: one first-time traveler was initially scored 35 (confidence 0.4), while two historical reports received 20 and 15. Later analysis of the full indexed set placed planning questions at approximately 35-65 and retrospective trip reports around 10-20. These are model outputs, not ground-truth measures of lead quality.

What was not achieved

There is no evidence of a real Reddit comment successfully published by the trusted workflow; qualified leads and confirmed external publications remained zero. The final Hallmark frontend and isolated verified backend were not shown merged into one independently inspected release. The multi-agent setup was not compared against a solo-agent baseline.

05 Failure modes and observations

01 Shared communication is not enforced ownership

The agents agreed on boundaries but later edited overlapping server files. Written coordination lacked a technical enforcement mechanism.

02 Passing tests can coexist with misleading UI

The frontend displayed Connected status without execution proof, counted imported excerpts as live verified threads, and sometimes invented absent source dates/authors.

03 Model-generated success is not external evidence

The Zernio adapter could fabricate comment IDs and account states. Reliable sign-off required upstream provider evidence or independent platform confirmation.

04 Provider integration changes architecture

An Apify actor exceeded the original HTTP timeout, prompting durable background jobs and polling rather than interpreting a timeout as zero discovered leads.

05 Contract drift happens quickly

Import envelopes, error fields, analytics keys, generation payloads and lead mutation shapes diverged as agents built separately.

06 Data provenance drives trust

Codex quarantined a smoke-test record rather than silently mixing it with genuine indexed results. Unknown dates and authors were meant to remain unknown.

07 Independent review became the stabilizer

Cross-agent checks exposed both code issues and unsupported product claims, though corrections became costly when they touched the same files.

08 Tool integration needs secret boundaries

Lovable declined API-key export; the eventual RSA-signed endpoint reused hosted capability while keeping the secret remote.

06 Lessons for multi-agent engineering

The minimum viable coordination stack

Communication layer: an append-only conversation.md for decisions, requests, handoffs and evidence pointers. Execution layer: separate Git branches/worktrees or strict path isolation. Contract layer: a single typed API schema with generated consumers. Verification layer: regression tests, browser acceptance tests and provider-confirmed side effects. Integration owner: one explicit party authorized to approve merges.

A practical protocol v2

Control	Concrete change
Task identity	Every work item gets task ID, owner, reviewer, status and acceptance test.
Handoff discipline	State the exact commit/branch, API change, assumptions and blockers.
Edit isolation	Use separate worktrees by default; no cross-owned edits without transfer.
Evidence grading	Label proposed, implemented, locally tested, provider verified and independently confirmed.
Truthful UI	Never promote configured to connected, drafted to published, or indexed to fully read.
Automated gate	Run build, typecheck, contract tests, regression tests and browser E2E before merge.
Review cadence	Require ACKs for blocking messages; escalate stale dependencies to integration owner.

When this approach is useful

This lightweight pattern looks well suited to short POCs where the work can be separated along stable interfaces, the developers share filesystem access, and a human integrator can review claims. It becomes less attractive when agents continually modify the same subsystem, write to one shared runtime store, or perform externally consequential actions without verification.

What to measure next

Repeat the same frozen specification under three conditions: Antigravity alone, Codex alone and both together. Hold acceptance tests constant. Measure time to passed acceptance, end-to-end correctness, provider/model cost, human interventions, regressions, coordination latency and quality. This is necessary before claiming faster delivery or lower cost.

07 Final assessment and next steps

Supported conclusion: Independent coding products can exchange actionable engineering information through a shared file, complete distinct components, identify each other's defects and jointly reach meaningful real-provider workflows. The experiment offers a credible feasibility demonstration of a lightweight cross-tool collaboration mechanism.

Equally important counterfinding: Shared Markdown did not stop ownership violations, simulated success, contract drift or breaking concurrent edits. Reliability improved when one agent segregated a tested implementation and insisted on provider-backed outcomes. The orchestration idea is viable as a starting point, but cannot replace engineering controls.

Recommended completion gates

- Merge the verified backend and approved Hallmark frontend in one reviewed integration branch.
- Run a fixed browser end-to-end script: authentic login, discover, inspect snippet, analyze, generate, edit, approve and save evidence of a manual action.
- Keep unverified Zernio publishing disabled until account scopes and actual Reddit comment ID are confirmed with the provider and source platform.
- Rotate any exposed OpenAI key and audit .env.example plus Git history before public demo.
- Tag every dataset, AI output, draft and lead stage with real provenance and verified timestamps.
- Only then conduct the controlled solo-versus-dual-agent benchmark.

Research takeaway

The hard part of multi-agent coding was not getting two agents to write code. It was getting them to produce one system whose claims could be trusted.

Evidence note

Primary materials: successive coordination-log snapshots from 13 October 2026 (18:31 through approximately 20:30 PKT), together with the operator's learnings.md journal. The earliest learnings.md stopped at 19:16; later events were reconstructed from the subsequent supplied coordination logs. This report does not claim external independent reproduction, industry-standard benchmarks, exact agent cost, or a finished production deployment.

Prepared for professional discussion and transparent sharing. Technical identifiers, account secrets and personal information were deliberately omitted from the public report.