Plan reference: `docs/strategy/EXECUTION_AND_MARKETING_PLAN_2026_08_17.md` §3 M-5 (Aug 19).

What this is

FinAdvantage ships a native MCP (Model Context Protocol) server that exposes our 120 finance tools to any MCP-compatible client — Claude, ChatGPT, Gemini, Cursor, or your own internal agent. The server runs on the same Railway service (`finadvantage-agents`) that powers the customer-facing product; there is no separate authentication surface, no separate billing surface, and no model lock-in.
If a customer prefers to drive the pipeline themselves — without our UI — they can. The same tools the FinAdvantage UI calls, they call. The same outputs the UI renders, they get.

Endpoint

- **Production:** `https://finadvantage-mcp-server-production.up.railway.app/mcp` (internal — auth required)
- **Auth:** Bearer token from `GET /api/integrations/oauth/start` (the same OAuth flow the UI uses); token-scoped to a single `(org_id, source_id)` install
- **Scope:** Read-only by default; write operations require an explicit `write_scope: true` in the request payload

What you can do with it

The 120 tools cover the full finance-toolkit surface:

Tier	Tools	Use case
T1 — SQL execution	`run_sql`	Run SQL against a DuckDB session loaded from uploaded workbooks
T2 — Matching	`composite_match`, `filter_split`, `fuzzy_match`, `exact_match`	Reconcile two data sources
T3 — Ingest	`parse_spreadsheet`, `extract_document`, `ocr_pdf`	Bring non-tabular sources in
T4 — Egress	`generate_excel_formatted`, `write_multi_sheet_excel`	Produce deliverables
T5 — External	`http_request`, `quickbooks_*`, `xero_*`	Talk to a customer's ERP
T6 — Semantic	`summarize_document`, `classify_document_type`	Per-row LLM work

The full tool registry is at `agents/src/workflow/tools/tool_registry.py`. New tools are added at `_initialize_non_atomic_tools()` per the same path PR #304 used for the reconciliation tools.

How to try it in 5 minutes

Why MCP, why now

The Aug 19 research across 12 peers (`/tmp/positioning_branding_research.json`) found Last Accounting Company (LAC, lastaccountingcompany.com) leads their pitch with *"a full accounting team, one MCP server away"* — same wedge we sit in, but they're claiming MCP-first positioning publicly and we weren't. We already ship the server (`agents/mcp_server/main.py`); we just weren't marketing it. This page + the hero copy + the sales one-pager are the three surfaces that close that gap.

What this server does NOT do (and why)

- **It does NOT replace the FinAdvantage UI.** The UI is what most customers want; MCP is for the customer who has a custom agent in production and wants to drop in ours as a tool. Both surfaces ship. The MCP server does not have a UI; it speaks JSON.
- **It does NOT expose raw customer data without a scope token.** Every tool call requires a `(org_id, source_id)` install. Cross-tenant reads are impossible by design — see `agents/src/services/signal_ingestion/rbac.py` for the model.
- **It does NOT bypass the audit trail.** Every tool call is logged in `reliability_events` (Stage A through K depending on tool kind) the same way UI-initiated calls are. The audit-grade traceability we publish for the UI applies equally to MCP-initiated calls.

Implementation references (for the engineering team)

- **Server entry:** `agents/mcp_server/main.py`
- **Tool dispatcher:** `agents/src/workflow/tools/tool_registry.py` — all 120 tools
- **Auth:** Bearer token from `backend/src/routes/integrations.ts`
- **Audit:** `agents/src/services/reliability_events.py`
- **Plan section:** §3 M-5 (Aug 19, added in the LAC-borrowed second tagline update)