Zemoso

CASE STUDY

How MagicBI built an AI-Native Analytics Platform with Anthropic's Claude Code and MCP

Introduction

MagicBI is an AI-native autonomous analytics and business intelligence platform designed to eliminate the analytics bottleneck that separates raw data from business-ready answers, helping organizations move from business questions and data toward decision-ready insights and narratives.

In enterprise BI, data accuracy and governance are non-negotiable—if an executive query generates an incorrect SQL join or misinterprets a metric definition, trust in the platform collapses.

To deliver this platform without compromising velocity, the engineering team adopted an AI-native software delivery lifecycle. By embedding Anthropic's Claude Code CLI and Model Context Protocol (MCP) across product specification, design implementation, development, testing, and security workflows, the team accelerated execution while maintaining strict architectural standards.

Zemoso partnered with MagicBI to design and orchestrate these Claude-powered workflows across the build.

Challenge

Building a modern analytics platform requires balancing two competing demands: giving data and business teams appropriate control over enterprise definitions, context, and governance, while enabling business consumers to interact with analytics using natural language.

Enterprise analytics systems can struggle when translating complex data structures and business definitions into understandable answers. Traditional setups often require manual SQL engineering or rigid reporting templates, creating bottlenecks for business teams while overburdening data teams.

Partnership challenge

As MagicBI scaled across a sophisticated data and application architecture, several standard developer friction points threatened execution speed:

- **Spec and design drift:** Translating product requirements from Notion and visual designs from Figma into production code could introduce subtle logic gaps and styling mismatches.
- **Multi-system complexity:** Keeping dependencies and data-related changes synchronized across systems required careful engineering oversight.
- **Access-control hardening:** Testing authorization behavior across multi-tenant environments required extensive edge-case validation.

> "We've been working with MagicBI on an AI-native engagement — one that leverages AI deeply both in how we build the product and in the intelligence it delivers. Enterprise data analytics is complex: how you ground data, layer in context, and make sure the output holds up when someone questions it. Claude has been genuinely useful in working through that complexity fast and build this grounds-up product."
>
> Kanchuki Sharma, **Head of Product and Design, Zemoso**

Engineering approach

The team integrated Anthropic's Claude Code and MCP directly into four core operational workstreams:

1. **Product specifications with Notion MCP**
   To reduce requirement misunderstandings, the team connected Claude Code to Notion through MCP. Developers adopted a structured behavior-driven development (BDD) loop in which product requirements were translated into Gherkin-style specifications (Given/When/Then). These specifications provided shared context for implementation and validation before pull requests.
2. **Terminal-native development and pair programming**
   Developers operated Claude Code inside local terminal workflows to reduce context switching. The team used Claude Code to understand the codebase, map package dependencies, implement production logic, and execute refactoring tasks directly against the local codebase.
3. **Direct visual design handoffs via Figma MCP**
   Front-end developers queried Figma designs directly inside local terminal workflows via MCP. Claude Code extracted layout properties, visual tokens, and responsive component logic directly into production code, bypassing manual UI translation steps.
4. **Migration orchestration and security testing**
   The team leveraged Claude Code to assist with migration work across multiple systems and to identify and test security edge cases. For security, Claude Code helped simulate authorization and permission-boundary scenarios, broadening test coverage while reducing manual effort.

Claude-powered intelligence inside MagicBI

Beyond the development lifecycle, Claude models power several AI capabilities within MagicBI. Claude helps interpret natural-language business questions, reason over relevant business and data context, orchestrate analytical workflows, and transform analytical results into decision-ready experiences:

1. **Intelligent Agent Orchestration**
   Claude helps interpret user intent and coordinate specialized analytical and reasoning workflows.
2. **Context-Aware Analytics**
   Rather than generating responses from natural language alone, Claude works with MagicBI's contextual knowledge intelligence layer to incorporate relevant business definitions, data relationships, and access policies.
3. **Structured Output**
   Claude helps transform complex computed analytical results into understandable, decision-ready experiences such as interactive charts, executive summaries, memos and structured decks, while deterministic analytical processing remains responsible for numerical correctness.

A disciplined approach to AI and analytical correctness

MagicBI uses generative AI for reasoning, interpretation, and communication while keeping analytical computation deterministic. This separation allows Claude to help users explore, understand, and communicate analytical results without making the language model responsible for numerical correctness.

Impact

Integrating Claude Code and MCP into a continuous engineering pipeline yielded clear operational results across key engineering benchmarks:

Workstream	Traditional baseline	Claude-native SDLC	Observed Impact
Developer onboarding	Multi-week architecture mapping and documentation walkthroughs	Context-grounded codebase mapping via Claude CLI	Reduced to 3 days*
Spec synchronization	Manual interpretation of PRDs; frequent revision cycles	Executable Gherkin specs via Notion MCP	100% spec alignment from design to PR*
Design translation	Manual front-end recreation from visual Figma comps	Direct visual-token extraction via Figma MCP	~60% reduction in front-end handoff time*
Schema and security verification	Hand-crafted migration scripts; reactive permission testing	AI-assisted migration workflows and simulated authorization tests	Same-day migration turnarounds*

* Impact figures are based on observed experience across selected workflows and are not independently audited benchmarks.

Why the partnership matters

MagicBI and Zemoso's collaboration demonstrates how Claude can support both sides of an AI-native product: the way sophisticated software is engineered and the intelligence experiences delivered to users.

For the engineering team, Claude Code and MCP provide contextual assistance across product specifications, design implementation, development, testing, and security. Within MagicBI, Claude helps power AI-driven experiences that allow business users to interact with analytics using natural language and receive decision-ready outputs.

This creates a continuous connection between AI-assisted development and AI-powered product experiences, enabling the team to move faster while maintaining the reliability, governance, and analytical rigor expected from enterprise software.

Leadership perspective

Hardik Chheda,
Founder & CEO
MagicBI.ai

> We wanted AI to fundamentally change not just how our customers interact with analytics, but how we build the product itself. Working with Zemoso and using Claude Code and MCP has helped us bring that philosophy into our engineering workflow while using Claude to power key intelligence experiences within MagicBI. The result is an AI-native approach to both building and using analytics.

Satish Madhira,
CEO
Zemoso

> MagicBI gave us an opportunity to apply Claude Code and MCP across a complex AI-native product lifecycle, going well beyond code generation. Together, we integrated Claude into product specification, development, design implementation, testing, and security workflows, helping the team move faster while maintaining the engineering rigor required for an enterprise analytics platform.

Conclusion

Shipping complex data systems requires an engineering workflow capable of matching the domain's complexity. By combining Claude-powered development workflows with a disciplined architecture for AI-assisted analytics, MagicBI and Zemoso have accelerated product development while maintaining the standards required for enterprise analytics.

Claude is now embedded not only in how the team builds the product, but also in how users interact with it.

About MagicBI

MagicBI is an AI-native autonomous analytics platform that gives business teams boardroom-ready answers, dashboards, and narratives directly from their data, without requiring SQL.

Proprietary algorithms, underlying source code, and specific enterprise client data remain protected in accordance with applicable non-disclosure and intellectual property agreements.

P.S. Jointly authored by Zemoso Technologies and MagicBI.

Related resources

Case Study
Scaling Insights:A Business Intelligence Solutionfor Franchise E...

Case Study
Digitally TransformingResource Allocationfor a Data An...

Blog
AI observability:Choosing the right toolfor produc...

To hire your custom product pod, write to sales@zemosolabs.com today!

© 2026 Zemoso Technologies