

The text-first stack: headless, composable, agent-native

Why the next generation of marketing technology runs on language, APIs and the tools you already own.

Executive summary

- Marketing teams are overloaded with platforms, dashboards and logins. Each new tool adds context switching and cognitive load.
- Headless, composable software separates the engine that does the work from the screens that show it, so results can flow into any tool.
- AI agents work in text. Software that exposes its functions as commands, APIs and MCP endpoints can be operated by agents and people alike.
- The result is fewer platforms and higher throughput: one engine, a chat or command line to drive it, and existing tools to display it.

The problem: dashboard sprawl

The typical martech stack grew one point solution at a time. Each platform arrived with its own interface, its own data model and its own learning curve. Marketers now spend real hours moving between screens, exporting and re-importing data, and reconciling numbers that disagree.

Adding another platform with another dashboard rarely fixes this. It adds one more place to look.

Headless and composable

Headless software keeps the engine separate from any front end. Composable architecture, often described by the MACH principles of microservices, API-first, cloud-native and headless, builds a stack from interchangeable parts that talk through APIs.

For marketing, the engine that finds leads, scores them, personalizes content and runs automations no longer needs its own dashboard. Its output can land in whatever the team already uses.

"If you think about it, [business applications] are essentially CRUD databases with a bunch of business logic."

Satya Nadella on the BG2 podcast, December 2024

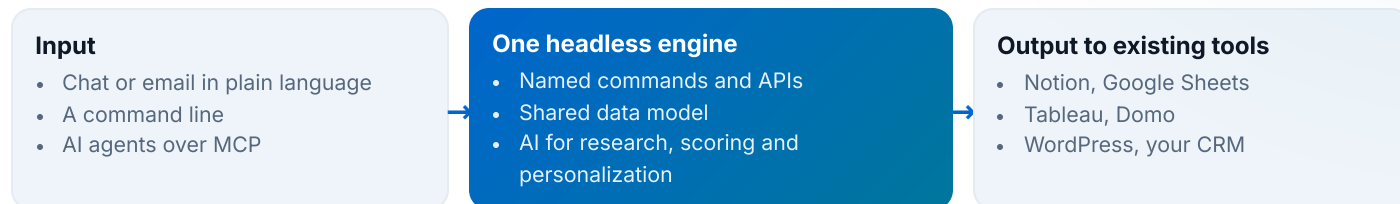
Agent-native software

AI agents do not click through interfaces. They read and write text, call functions and chain steps together. Software becomes agent-native when its capabilities are exposed as clearly named actions, through APIs and through the Model Context Protocol (MCP), an open standard for connecting AI models to tools and data.

Microsoft CEO Satya Nadella expects SaaS and business applications to collapse in the agent era. His explanation: "If you think about it, [business applications] are essentially CRUD databases with a bunch of business logic." As that logic moves into agents that work across every system, software gains a vocabulary, and each capability becomes a word an agent or a person can use.

Text as the interface

Developers have long preferred command lines for speed. Many major platforms, including Stripe, GitHub and Vercel, ship a command line tool beside their web dashboard. One typed command can replace a dozen clicks, it chains into the next command, repeats exactly, and leaves a readable record of what happened. Agents bring the same advantages to people who never write code: they type or say what they want, and the agent runs the commands.



Consolidate instead of adding

A lean, text-driven engine with an AI layer does not need a new platform around it. It writes results into the reporting, documentation and content tools a team already owns, and those views update as the engine works. The team gets new capability while the number of platforms stays flat or shrinks.

Keep a human surface

An engine operated only by agents has no surface a person can inspect. A command line keeps one. People can see every action an agent took, repeat it, or do the work by hand. Libraries such as cmd2 and Typer in Python and Cobra in Go make these interfaces quick to build, and plugin patterns like those used by git and kubectl let new commands drop in without touching the core.

Where it gets harder

Command lines are hard to browse, so good help, auto-completion and a natural language layer matter. Many people still prefer to click, which is why output belongs in familiar tools rather than in yet another interface. And every part of the engine must share one definition of a company, a contact and a campaign, or fragmentation returns in a new form.

Questions to ask of any new martech

- Does it expose its functions through APIs or MCP?
- Can its output land in the tools we already use?
- Can an agent operate it, and can a person see what the agent did?
- Does it reduce the number of platforms we run, or add one?

About Nathan T Warne

Nathan T Warne is a marketing technology and automation consultant who helps organizations, agencies, and technology specialists build modern martech solutions. He brings more than 15 years in marketing for B2B technology companies, technology distributors, and marketing agencies. His hyper-personalization platform has been in production since 2025, sold through a partner agency to a global managed service provider and an industry-leading data platform company.

Ready to simplify your martech stack?

[Book a strategy session](#)

nathantwarne.com