Natoma Agent Access

For Secure Code Generation Workflows with MCP-enabled IDEs

AI-powered code generation tools like **Cursor** and **Windsurf** are transforming software development by increasing velocity and reducing manual effort. These tools enhance productivity by embedding intelligent assistants directly into the development workflow —from smart code suggestions to automated testing, debugging, and refactoring.

Model Context Protocol (MCP) takes this further by enabling IDEs to interact programmatically with a wide range of developer tools and enterprise tools. This unlocks advanced capabilities such as real-time codebase awareness, secure document access, tool chaining, and context-rich model outputs, all without disrupting developer productivity.

However, realizing the full potential of these tools in enterprise environments depends on secure, governed access to internal knowledge—including code, documentation, and tool context. **Natoma** enables organizations to provision this knowledge intelligently, ensuring AI assistants are effective without compromising on security, compliance, or governance at scale.

The Challenges

Adopting MCP-enabled IDEs offers enormous productivity gains—but only when paired with the right security, governance, and integration practices.

1. **Local Hosting Complexity**
   Many developers face difficulties setting up and maintaining local MCP servers. The operational burden reduces accessibility and introduces inconsistency in usage across teams

2. **Lack of Visibility & Monitoring**
   MCP enables IDEs to connect to any enterprise data source, but security teams often lack oversight into which endpoints or data systems are being accessed, leading to shadow AI.

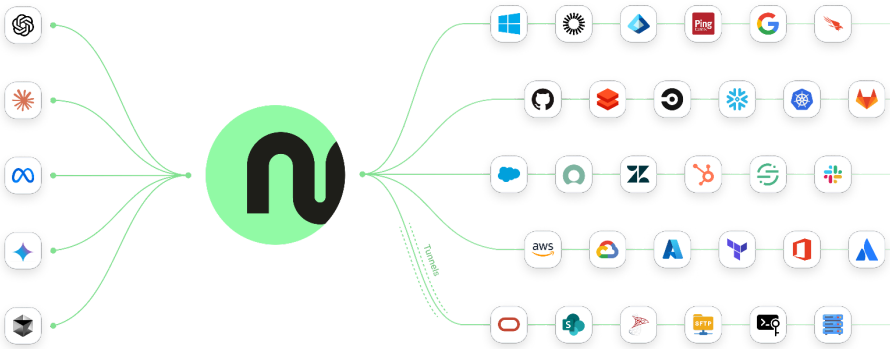
3. **Unmanaged MCP Servers**
   Without strict controls & governance, developers may connect to unvetted, unverified, and unsanctioned servers, which increases the risk of:

   - Tool positioning attacks
   - Context leakage
   - Tool hijacking
   - Indirect server hijacking
   - Tools shadowing

4. **Excessive access permissions**
   Over-permissioned access through unmanaged or unverified MCP servers can open doors to data leakage, exfiltration & misuse, lateral movement, or escalation attacks.

The Natoma Edge

Natoma Agent Access provides the fastest and most secure way to operationalize AI-powered development workflows—without increasing complexity or risk. Natoma's **hosted MCP platform** offers plug-and-play integration that works out of the box with tools like Cursor, Windsurf, Copilot, and internal chat agents.



Key Principles for Secure Adoption

Secure adoption starts with structure, control, and visibility—Natoma Agent Access delivers all three by connecting IDEs and AI agents to enterprise tools and data through a governed, policy-driven framework.

- **Security-first architecture** – Restricts AI agents to verified MCP servers, prevents shadow access, and ensures safe tool chaining.
- **Fine-grained authorization** - Maintain control over what end users can see and do based on their roles and responsibilities.
- **Granular context governance** – Automatically redacts sensitive content, classifies internal docs, and enforces policy-driven access based on role and function.
- **End-to-end traceability** – Every interaction is logged and auditable, with built-in policy controls, anomaly detection, and reporting for compliance teams.
- **Enterprise-scale deployment** – Flexibility to run on your cloud and seamlessly integrates with systems like Okta, GitHub, Confluence, and Notion, requiring minimal lift from engineering or security teams.

Natoma delivers simplicity, scalability, and security—so teams can move faster while staying in control.

For more information, visit mcp.natoma.id or connect with our directly team at natoma.id/book-a-demo.