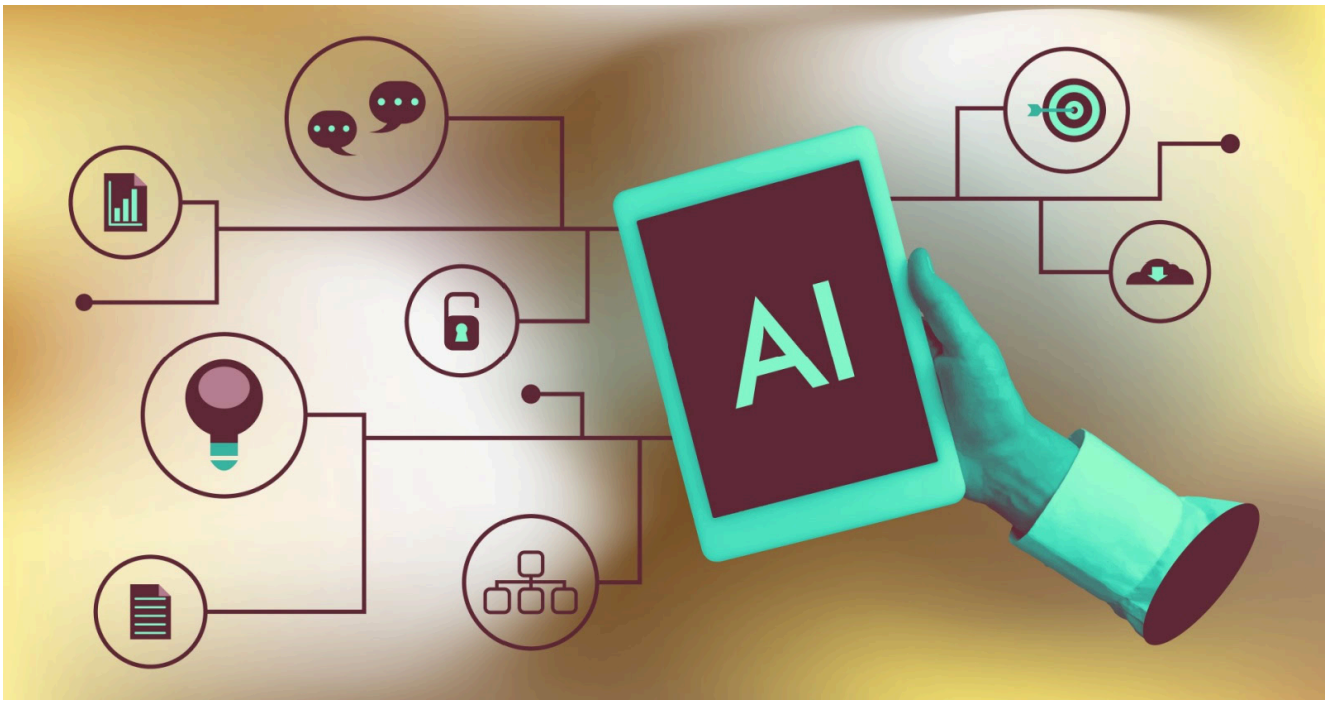
MCP, A2A, NLWeb, And AGENTS.md: The Standards Powering The Agentic Web via @sejournal, @slobodanmanic



This is Part 3 in a five-part series on optimizing websites for the agentic web. Part 1 covered the evolution from SEO to AAIO. Part 2 explored how to get your content cited in AI responses. This article goes deeper: the protocols forming the infrastructure layer that make everything else possible.

The early web needed HTTP to transport data, HTML to structure content, and the W3C to keep everyone building on the same foundation. Without those shared standards, we'd have ended up with a fragmented collection of incompatible networks instead of a single web.

The agentic web is at that same inflection point. AI agents need standardized ways to connect to tools, talk to each other, query websites, and understand codebases. Without shared protocols, every AI vendor builds proprietary

integrations, and the result is the same fragmentation the early web narrowly avoided.

Four protocols are emerging as the foundational layer. This article covers what each one does, who’s behind it, and what it means for your business. Throughout this series, we draw exclusively from official documentation, research papers, and announcements from the companies building this infrastructure.

Why Standards Matter

Consider how the original web came together. In the early 1990s, competing browser vendors and incompatible standards were fragmenting what should have been a unified network. The W3C brought order by establishing shared protocols. HTTP handled transport. HTML handled structure. Everyone agreed on the rules, and the web took off.

AI is at a similar crossroads. Right now, every major AI company is building agents that need to interact with external tools, data sources, other agents, and websites. Without standards, connecting your business systems to AI means building separate integrations for Claude, ChatGPT, Gemini, Copilot, and whatever comes next. That’s the M x N problem: M different AI models times N different tools equals an unsustainable number of custom connections.

What makes this moment remarkable is who’s building the solution together. On Dec. 9, 2025, the Linux Foundation announced the Agentic AI Foundation (AAIF), a vendor-neutral governance body for agentic AI standards. Eight platinum members anchor it: AWS, Anthropic, Block, Bloomberg, Cloudflare, Google, Microsoft, and OpenAI.

OpenAI, Anthropic, Google, and Microsoft. Competing on AI products, collaborating on AI infrastructure. As Linux Foundation Executive Director Jim Zemlin put it: “We are seeing AI enter a new phase, as conversational systems shift

to autonomous agents that can work together."

This is a bigger deal than most people realize. Competitors building shared infrastructure because they all recognize that proprietary standards would hold back the entire ecosystem, including themselves.

MCP: The Universal Adapter

What it is: The Model Context Protocol (MCP) is an open standard for connecting AI applications to external tools, data sources, and workflows.

The official analogy is apt:

> *"Think of MCP like a USB-C port for AI applications. Just as USB-C provides a standardized way to connect electronic devices, MCP provides a standardized way to connect AI applications to external systems."*

Before MCP, if you wanted your database, CRM, or internal tools accessible to an AI assistant, you had to build a custom integration for each AI platform. MCP replaces that with a single standard interface. Build one MCP server for your data, and every MCP-compatible AI system can connect to it.

The numbers are striking. MCP launched as an open-source project from Anthropic on Nov. 25, 2024. In just over a year, it reached 97 million monthly SDK downloads across Python and TypeScript, with over 10,000 public MCP servers built by the community.

The adoption timeline tells the story. Anthropic's Claude had native MCP support from day one. In March 2025, OpenAI CEO Sam Altman announced support across OpenAI's products, stating: "People love MCP and we are excited to add support across our products." Google followed in April, confirming MCP support in Gemini. Microsoft joined the MCP steering committee at Build

2025 in May, with MCP support in VS Code reaching general availability in July 2025.

From internal experiment to industry standard in 12 months. That pace of adoption signals something real.

What this means for your business: If your data, tools, or services are MCP-accessible, every major AI platform can use them. That's not a theoretical benefit. It means an AI assistant helping your customer can pull real-time product availability from your inventory system, check order status from your CRM, or retrieve pricing from your database, all through one standardized connection rather than platform-specific integrations.

A2A: How Agents Talk To Each Other

What it is: The Agent2Agent protocol (A2A) enables AI agents from different vendors to discover each other's capabilities and collaborate on tasks.

If MCP is how agents connect to tools, A2A is how agents connect to each other. The distinction matters. In a world where businesses use AI agents from Salesforce for CRM, ServiceNow for IT, and an internal agent for billing, these agents need a way to discover what each other can do, delegate tasks, and coordinate responses. A2A provides that.

Google launched A2A on April 9, 2025 with over 50 technology partners. By June, Google donated the protocol to the Linux Foundation. By July, version 0.3 shipped with over 150 supporting organizations, including Salesforce, SAP, ServiceNow, PayPal, Atlassian, Microsoft, and AWS.

The core concept is the **Agent Card**: a JSON metadata document that serves as a digital business card for agents. Each A2A-compatible agent publishes an Agent Card at a standard web address (/.well-known/agent-card.json) describing its

identity, capabilities, skills, and authentication requirements. When one agent needs help with a task, it reads another agent's card to understand what that agent can do, then communicates through A2A to request collaboration.

Google's own framing of how these pieces fit together is useful: "Build with ADK, equip with MCP, communicate with A2A." ADK (Agent Development Kit) is Google's framework for building agents, MCP gives them access to tools, and A2A lets them talk to other agents.

Here's a practical example. A customer contacts your company with a billing question that requires a refund. Your customer service agent (built on one platform) identifies the issue, passes the context to your billing agent (built on another platform) via A2A, which calculates the refund amount and hands off to your payments agent (yet another platform) to process it. The customer sees one seamless interaction. Behind the scenes, three agents from different vendors collaborated through a shared protocol.

The enterprise adoption signal is strong. When Salesforce, SAP, ServiceNow, and every major consultancy sign on to a protocol within months, it's because their enterprise clients are already running into the multi-vendor agent coordination problem that A2A solves.

NLWeb: Making Websites Conversational

What it is: NLWeb (Natural Language Web) is an open project from Microsoft that turns any website into a natural language interface, queryable by both humans and AI agents.

Of the four protocols covered here, NLWeb is the most directly relevant to this series' audience. MCP, A2A, and AGENTS.md are primarily developer concerns. NLWeb is about your website.

NLWeb was introduced at Microsoft Build 2025 on May 19, 2025. It was conceived and developed by R.V. Guha, who joined Microsoft as CVP and Technical Fellow. If that name sounds familiar, it should: Guha is the creator of RSS, RDF, and Schema.org, three standards that fundamentally shaped how the web organizes and syndicates information. When the person behind Schema.org builds a new web protocol, it's worth paying attention.

The key insight behind NLWeb is that websites already publish structured data. Schema.org markup, RSS feeds, product catalogs, recipe databases. NLWeb leverages these existing formats, combining them with AI to let users and agents query a website's content using natural language instead of clicking through pages.

Microsoft's framing is deliberate: "NLWeb can play a similar role to HTML in the emerging agentic web." The NLWeb README puts it even more directly: "NLWeb is to MCP/A2A what HTML is to HTTP."

Every NLWeb instance is automatically an MCP server. That means any website running NLWeb immediately becomes accessible to the entire ecosystem of MCP-compatible AI assistants and agents. Your website's content doesn't just sit there waiting for visitors. It becomes actively queryable by any AI system that speaks MCP.

Early adopters include Eventbrite, Shopify, Tripadvisor, O'Reilly Media, Common Sense Media, and Hearst. These are content-rich websites that already invest heavily in structured data. NLWeb builds directly on that investment.

Here's what this looks like in practice. Instead of a user navigating Tripadvisor's search filters to find family-friendly restaurants in Barcelona with outdoor seating, an AI agent could query Tripadvisor's NLWeb endpoint: "Find family-friendly restaurants in Barcelona with outdoor seating and

good reviews." The response comes back as structured Schema.org JSON, ready for the agent to present to the user or act on.

If your business has already invested in Schema.org markup (and Part 2 of this series explained why you should), you're closer to NLWeb readiness than you might think.

AGENTS.md: Instructions For AI Coders

What it is: AGENTS.md is a standardized Markdown file that provides AI coding agents with project-specific guidance, essentially a README written for machines instead of humans.

This protocol is less directly relevant to the marketers and strategists reading this series, but it's an important piece of the complete picture, especially if your organization has development teams using AI coding tools.

AGENTS.md emerged from a collaboration between OpenAI Codex, Google Jules, Cursor, Amp, and Factory. The problem they were solving: AI coding agents need to understand project conventions, build steps, testing requirements, and architectural decisions before they can contribute useful code. Without explicit guidance, agents make assumptions that lead to inconsistent, buggy output.

Since its release in August 2025, AGENTS.md has been adopted by over 60,000 open-source projects and is supported by tools including GitHub Copilot, Claude Code, Cursor, Gemini CLI, VS Code, and many others. It's now governed by the Agentic AI Foundation, alongside MCP.

The file itself is simple. Plain Markdown, typically under 150 lines, covering build commands, architectural overview, coding conventions, and testing requirements. Agents read it before making any changes, getting the same tribal knowledge that

senior engineers carry in their heads.

GitHub reports that Copilot now generates 46% of code for its users. When nearly half of code is AI-generated, having a standard way to ensure agents follow your conventions, security practices, and architectural patterns isn't optional. It's quality control.

Why this matters for your business: If your development teams use AI coding tools (and most do), AGENTS.md ensures those tools produce code that matches your standards. It reduces agent-generated bugs, cuts onboarding time for AI tools on new projects, and provides consistency across teams.

How They Fit Together

These four protocols aren't competing. They're complementary layers in the same stack.

Protocol	Created By	Purpose	Web Analogy
MCP	Anthropic	Connect agents to tools and data	USB ports
A2A	Google	Agent-to-agent communication	Email/messaging
NLWeb	Microsoft	Make websites queryable by agents	HTML
AGENTS.md	OpenAI + collaborators	Guide AI coding agents	README files
AAIF	Linux Foundation	Governance and standards body	W3C

The stack works like this: MCP provides the plumbing for agents to access tools and data. A2A enables agents to coordinate with each other. NLWeb makes website content accessible to the entire ecosystem. AGENTS.md ensures AI coding agents build correctly. And the Agentic AI Foundation provides the governance layer, ensuring these protocols remain

open, vendor-neutral, and interoperable.

The parallel to the original web is impossible to ignore:

- **HTTP** (transport) maps to **MCP** (tool access) and **A2A** (agent communication).
- **HTML** (content structure) maps to **NLWeb** (website content for agents).
- **W3C** (governance) maps to **AAIF** (governance).

What's different this time is the speed. HTTP took years to gain broad adoption. MCP went from launch to universal platform support in 12 months. A2A grew from 50 to 150+ partner organizations in three months. NLWeb shipped with major publisher adoption at launch. AGENTS.md reached 60,000 projects within its first few months.

The infrastructure is being built at internet speed, not standards-committee speed. That's partly because the companies involved are the same ones building the agents that need these protocols. They're motivated.

And these four aren't the only protocols emerging. Commerce-specific standards are building the transaction layer: Shopify and Google co-developed the Universal Commerce Protocol (UCP), launched in January 2026 with support from Etsy, Target, Walmart, and Wayfair. OpenAI and Stripe co-developed the Agentic Commerce Protocol (ACP), which powers Instant Checkout in ChatGPT. CopilotKit's AG-UI protocol addresses agent-to-frontend communication, with integrations from LangGraph, CrewAI, and Google ADK. We'll cover the commerce protocols in depth in Part 5.

What This Means For Your Business

You don't need to implement all four protocols tomorrow. But you need to understand what's being built, because it shapes what your website, tools, and teams should be ready for.

If you've already invested in Schema.org markup, NLWeb is your closest on-ramp. It builds directly on the structured data you already maintain. As NLWeb adoption grows, your Schema.org investment becomes the foundation for making your website conversationally accessible to AI agents. Keep your structured data current and comprehensive.

If you have APIs or internal tools, consider MCP accessibility. Making your services available through MCP means any AI platform can interact with them. For ecommerce, that could mean product catalogs, inventory systems, and order tracking becoming accessible to AI shopping assistants across ChatGPT, Claude, Gemini, and whatever comes next.

If you're evaluating multi-vendor agent workflows, A2A is the protocol to watch. Enterprise organizations running agents from multiple vendors (Salesforce, ServiceNow, internal tools) will increasingly need these agents to coordinate. A2A is the emerging standard for that coordination.

If your development teams use AI coding tools, adopt AGENTS.md now. It's the simplest protocol to implement (it's a single Markdown file) and the one with the most immediate, tangible benefit: fewer bugs, more consistent output, faster onboarding for AI tools on your codebase.

The underlying message across all four protocols is the same: the agentic web is being built on open standards, not proprietary ones. The companies that understand these standards early will be better positioned as AI agents become a primary way users interact with businesses.

These aren't things you need to implement today. But they are things you need to understand, because Part 4 of this series gets into the technical specifics of making your website agent-ready.

Key Takeaways

- **Four protocols form the agentic web's infrastructure.** MCP (tools), A2A (agent communication), NLWeb (website content), and AGENTS.md (code guidance) are complementary layers, not competitors.
- **The speed of adoption signals real urgency.** MCP reached 97 million monthly SDK downloads and universal platform support in 12 months. A2A grew from 50 to 150+ partner organizations in three months. These are not experiments.
- **Competitors are collaborating on infrastructure.** OpenAI, Anthropic, Google, and Microsoft are all building shared protocols under the Agentic AI Foundation. This mirrors the W3C moment that unified the early web.
- **NLWeb is potentially the most relevant protocol for website owners.** Built by the creator of Schema.org, it turns your existing structured data into a conversational interface for AI agents. Every NLWeb instance is automatically an MCP server.
- **MCP is the universal adapter.** Build one MCP connection to your data, and every major AI platform (Claude, ChatGPT, Gemini, Copilot) can access it. No more building separate integrations for each platform.
- **Start with what you have.** Schema.org markup readies you for NLWeb. Existing APIs can become MCP servers. AGENTS.md is a single file your dev team can create today. You don't need to start from scratch.

The original web succeeded because competitors agreed on shared standards. The agentic web is following the same playbook, just faster. The protocols are being established now. The governance is in place. The agents are already using them.

Up next in Part 4: the hands-on technical guide for making your website ready for autonomous AI agents, from semantic

HTML to accessibility standards to testing with real agent tools.

More Resources:

This post was originally published on No Hacks.

Featured Image: Collagery/Shutterstock

https://www.searchenginejournal.com/mcp-a2a-nlweb-and-agents-md-the-standards-powering-the-agentic-web/570092/