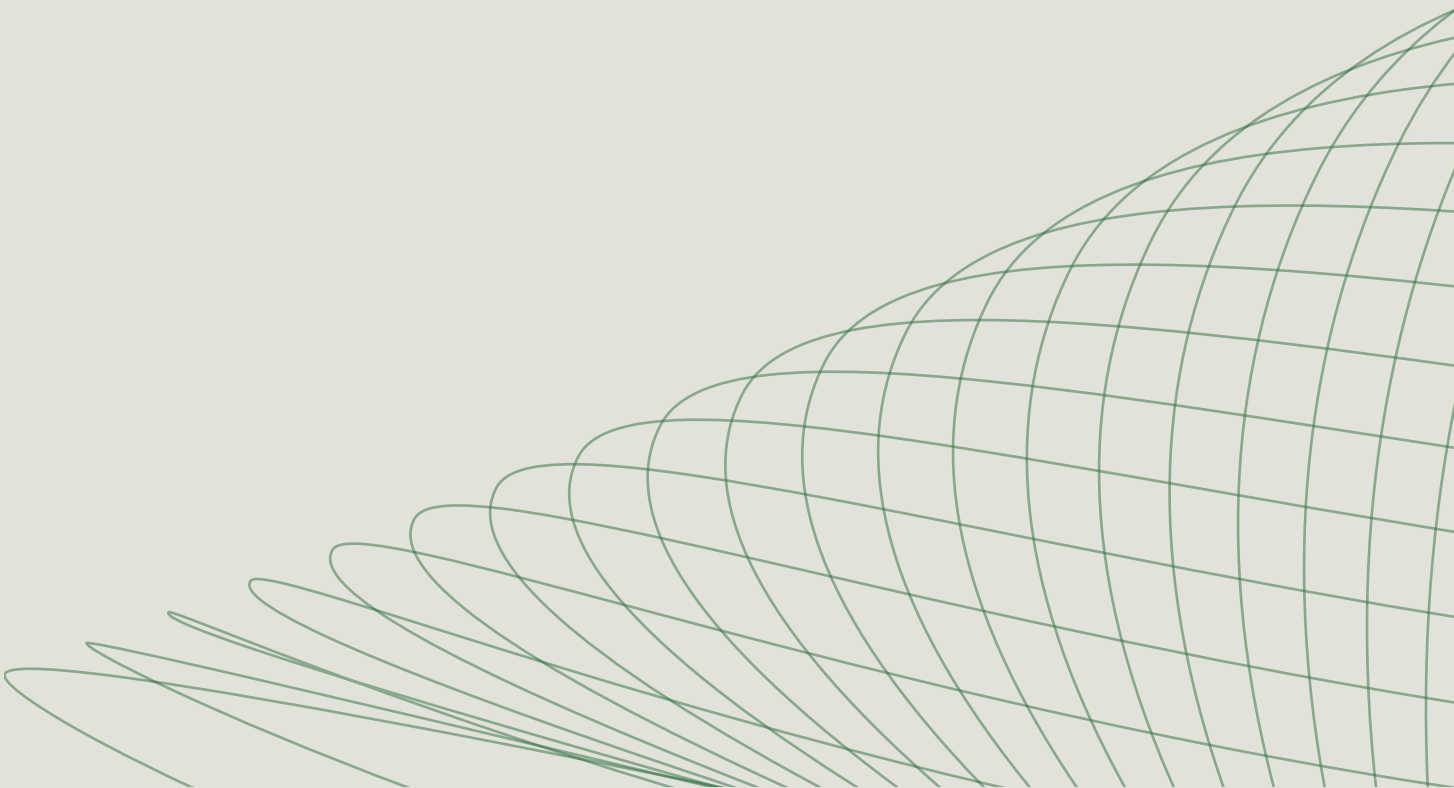
STACKLOK

MCP Maturity Model for Retail (2026)

Identify where your retail organization is today, and drive deeper adoption of AI agents and MCP servers



Contents

Introduction

The Model Context Protocol (MCP) represents a fundamental shift in how AI systems integrate with enterprise tools and data sources. Introduced by Anthropic in November 2024, MCP standardizes how AI systems connect to external services and data, extending the capabilities of AI models, agents and assistants.

MCP has evolved rapidly. Tens of thousands of MCP servers have been published by individual developers through to some of the world's largest enterprises. MCP has evolved from proposal to fully governed protocol with a formal structure, release cadence and roadmap.

While it is still early days, retail technology leaders are embracing MCP now because they recognize the opportunity for competitive advantage. In an industry defined by thin margins, demanding customers, and relentless operational complexity, they are multiplying the value of their AI agents and assistants, and moving pilots into production to drive measurable outcomes.

The MCP Imperative for Retail

There is another reason for retail leaders to embrace MCP: your employees are already using MCP servers, and that is creating risk for your business. Inside most retailers, MCP adoption is not a controlled, top-down initiative. With thousands of MCP servers now publicly available, it is easier than ever for an analyst or engineer to wire an AI agent directly into your product catalog, order management system, customer database, or supplier APIs. This creates a growing operational and security gap.

This is the shadow AI problem: unsanctioned MCP servers running on local machines, and unknown tools reaching into systems that hold customer PII, payment data, and commercially sensitive pricing information. Shadow AI emerges as people reach for whatever helps them move faster, and in retail, the pressure to move fast is constant. Unless you provide a safe, governed alternative, this usage continues invisibly and without oversight.

You can change the equation by standardizing the way your teams (and their AI agents and assistants) access context behind your corporate firewall. Rather than blocking MCP adoption (an approach that inevitably fails), retailers must offer a governed MCP platform that delivers the same speed and flexibility users want, but with enterprise-grade security, auditability, and control. The imperative is clear: adopt MCP intentionally, or inherit an ungoverned landscape of ad-hoc integrations that put customer data, compliance, and trading operations at risk.

MCP Before and After: A Retail View

A common question we receive from retailers at the earliest stages of the maturity model is, "What will we be able to do with MCP that we cannot do without it?"

Today, most AI agents and assistants show up like an intern. They are capable, but they know nothing about your business and so they require constant supervision to produce useful work. That is why roughly 46% of AI pilots are currently failing, and why so many leave end users and internal champions frustrated.

Now consider what it would be like to operate as an MCP-native retailer. Some concrete examples of what becomes possible:

- **Merchandising and supply chain teams** will have agents that monitor inventory positions across channels, flag stockouts and overstocks before they affect sales, propose replenishment and markdown actions for human approval, and update planning systems automatically, freeing planners to focus on strategy rather than spreadsheets.
- **Customer service and contact center agents** will know a shopper's entire history with your brand (every order, return, loyalty interaction, and prior contact) and will resolve queries with fully personalized solutions, massively increasing first-contact resolution and deflection rates.
- **Store and operations teams** will be able to query a central service that surfaces the exact product, pricing, promotion, or policy information they need in seconds (in accordance with their permissions), without navigating internal systems or waiting on head-office queues.
- **E-commerce and digital teams** will ship personalized experiences, dynamic merchandising, and customer-facing assistants faster, because the underlying context is governed, reusable, and available through a single interface.
- **All of the above** happens under your watch and within the full control of your private cloud. Every interaction aligns with your governance, data protection, and security policies.

In short, the promise of AI in retail has a massive dependency on context acquisition and optimization, and therefore, on MCP.

MCP FAQ

Before we get to the Maturity Model, let's address a few common questions. We assume you are reading this because you are interested in MCP, but we will not assume you have pushed in all your chips.

Why should I bet on MCP rather than other protocols like A2A (Google) or ACP (IBM)?

At present, MCP is the only protocol with real momentum. As of December 2025, more than 30,000 MCP servers had been published. Major providers that retailers rely on, including Salesforce, SAP, Shopify, GitHub, Atlassian, and others have introduced MCP servers but have not invested in competing protocols. MCP is the only real game in town.

Can't we avoid MCP by tuning agent .md files or enabling Anthropic's "skills"?

Tuning agents via .md files and skills are low-friction steps towards better performance, but neither gives an agent real, valuable context. A .md file can instruct an agent on your coding or content style, but it does not connect it to the product, customer, and order data behind your firewall. Skills provide specific executables, but they are not portable across agents or discoverable across your organization, so they serve individual users, not the enterprise.

What are the security risks of adopting MCP, given reports of tool poisoning and other attacks?

Security is a real consideration when you are connecting AI systems to customer data, payment systems, and supplier integrations. When the MCP spec was first introduced, the security approach was overly simplistic. Since then the spec has been revised to introduce OAuth, and enterprise MCP platforms have come to market with advanced security capabilities.

If you have other questions, please ask. We have an active Discord community at https://discord.gg/stacklok, or email enterprise@stacklok.com and we will respond quickly.

The Retail MCP Maturity Model

This Maturity Model gives retail organizations a framework to assess their current MCP adoption level and chart a path toward becoming an MCP-native retailer. We start with a summary of the four stages, with both technical and process markers for each:

MCP STAGE	Experiment & Prepare	Build Pilots & Capabilities	Scale MCP Infrastructure	MCP-Native Retailer
DEPLOYMENT	Local UI / CLI with containers	Local UI / CLI with containers	Local and K8s Operator	Service mesh
FOCUS	Discovery & learning	Pilots & deployment	Platform & governance	Innovation & revenue
PROCESS MARKERS	Individual analyst / dev experiments Ungoverned use Off-the-shelf integrations with current agents	Team-led pilots (e-comm, supply chain, CX) AI pilot success metrics defined Use-case-specific pilots	Enterprise-led deployments across banners Governance frameworks Context acquisition strategy	Enterprise-wide context infrastructure AI-embedded customer experiences Asynchronous, agentic retail workflows
TECHNICAL MARKERS	MCP servers on personal laptops No standardized connection practices No containerization, RBAC or audit trail Secrets stored locally / in plain text	MCP servers deployed in containers Standardized authentication Basic monitoring and logging Secrets managed via Vault / 1Password	MCP servers deployed via K8s Consistent RBAC across banners / namespaces Network policies enforced Observability stack integrated	Autonomous orchestration across servers Hybrid registries: public, private, SaaS Dynamic permissions & token exchange Policy engine evaluates actions

On the following pages, we examine each stage in more detail, including the risks of stalling, the metrics that signal readiness to advance, and the roles you will need in place.

Stage 1: Experiment & Prepare

Retailers at Stage 1 are just beginning to explore what MCP can do. Early adopters are usually developers, data analysts, or digital-team engineers experimenting with public MCP servers or simple local integrations to accelerate their work. Adoption is organic, uncoordinated, and largely invisible to IT and security, creating the first signs of shadow AI risk. This stage is defined by the realization that AI agents become dramatically more useful once connected to real retail systems and data.

Process Markers

- Individual analysts or engineers run MCP servers locally against systems like product catalogs, Google Analytics, or order data, without visibility
- Multiple AI tools are used inconsistently across merchandising, e-commerce, supply chain, and marketing teams
- No standard for authentication, secrets, or connectivity to commerce and customer systems
- Security learns about AI usage ad hoc, often after a data-handling concern is raised
- Early wins exist (a useful inventory query, a faster content draft) but are isolated and cannot be repeated across teams

Technical Markers

- MCP servers run on laptops or personal sandboxes
- Basic filesystem or public MCP servers used for experimentation
- No containerization, network isolation, RBAC, or audit trail
- Credentials to retail systems (CRM, OMS, analytics) stored locally instead of an enterprise vault
- No standardized connection patterns for AI tools

Risks if You Stay in This Stage

- Shadow AI usage expands faster than IT can understand or govern
- Customer PII, payment, or pricing data is accessed or moved without visibility, which is a PCI DSS and GDPR exposure
- Teams duplicate work because no shared registry or patterns exist (the same Salesforce or Shopify integration rebuilt five times)
- Early AI wins stall because they cannot scale into production

Success Metrics (to complete Stage 1)

- Inventory of all known MCP usage across retail teams
- At least 2–3 high-value MCP-assisted workflows identified (e.g. inventory lookups, customer-history summarization, content generation)
- Baseline security and data-handling requirements agreed with InfoSec and your data protection function
- Decision made on deployment path (local + containerization vs. centralized)

Required Roles

- Developer / analyst experimenting with MCP
- Security and data-protection partner providing initial review guidance
- Program sponsor or digital / AI innovation lead

Stage 2: Build Pilots & Capabilities

In Stage 2, retailers move beyond experimentation and begin deploying MCP servers for real workflows and early production use cases, such as a contact-center assistant, a merchandising copilot, or an internal knowledge agent. Teams start building custom MCP servers, standardizing connection patterns, and establishing security controls. Pilots demonstrate clear productivity gains, but each team is still operating semi-independently. This stage marks the transition from isolated experiments to repeatable, policy-aligned capabilities.

Process Markers

- Teams across e-commerce, supply chain, and customer experience are running MCP-based pilots for real workflows
- Early internal champions are emerging and sharing best practices
- Security has begun formalizing policies for AI access to customer and commercial data, but lacks automation
- Leadership recognizes MCP's potential and expects validated trading or productivity outcomes

Technical Markers

- MCP servers deployed in containers (Docker, Podman), not on laptops
- Standardized authentication established (OAuth/OIDC) against your IdP
- Reusable MCP configurations for dev/test environments
- Basic monitoring and logging instrumented (stdout, container logs)
- Secrets to retail systems managed through Vault, 1Password, or Kubernetes Secrets

Risks if You Stay in This Stage

- Pilots remain siloed by line of business or function and fail to converge into a platform strategy
- Security fatigue grows due to inconsistent policy implementation across teams handling customer data
- Operational burden increases as more pilot servers are added
- Fragmentation makes it harder to migrate later to centralized governance, especially across multiple lines of business

Success Metrics (to complete Stage 2)

- ≥ 5 pilot-grade MCP servers deployed with consistent container standards
- Reusable connection templates and permission profiles created (e.g. a read-only customer-lookup profile)
- MCP-assisted workflows producing measurable productivity or trading impact
- Ability to deploy a new MCP server in < 1 day (including configuration + testing)

Required Roles

- Platform engineer / DevOps practitioner
- Security architect (MCP permission profiles, secrets and data-handling policies)
- Pilot workflow owners (merchandising, supply chain, CX, digital)
- AI program manager tracking reuse and outcomes

Stage 3: Scale MCP Infrastructure

Stage 3 represents the shift from team-level pilots to an enterprise-wide platform. MCP servers are centrally orchestrated, governed, and deployed at scale through Kubernetes, CRDs, and automated policies. Multiple business units now rely on MCP-powered workflows, and the platform team is focused on reliability, observability, and security enforcement, including readiness for peak periods. MCP becomes a shared organizational capability rather than a niche tool.

Process Markers

- Multiple business units are requesting access to MCP-powered workflows
- Pressure increases to provide a governed, reliable, enterprise-wide platform
- AI assistants need MCP toolchains to support cross-system workflows (order + inventory + customer)
- Teams begin expecting production-grade SLAs for MCP availability, including during peak season

Technical Markers

- MCP servers orchestrated via Kubernetes Operator or similar automation
- RBAC applied consistently across namespaces, teams, and business units
- Network policies enforced; egress to customer and payment systems is controlled
- Centralized registry or catalog of MCP servers introduced
- Observability stack deployed (OpenTelemetry, Prometheus, log aggregation)
- GitOps used for MCP server CRDs and configuration updates

Risks if You Stay in This Stage

- Platform team becomes a bottleneck for MCP server deployment, slowing trading teams
- Security posture weakens as the number of servers touching customer data grows
- Banners or business units build their own shadow platforms if the central one stalls
- Operational cost and toil scale non-linearly due to manual processes

Success Metrics (to complete Stage 3)

- ≥ 70% of MCP servers deployed via centralized platform (Kubernetes operator or equivalent)
- Mean time to approve + deploy a new MCP server: 3–5 days
- Full audit log retention meeting PCI DSS, GDPR, and CCPA requirements
- ≥ 80% of internal API domains exposed via MCP where appropriate (catalog, orders, pricing, loyalty)
- Cross-team adoption: ≥ 5 business units using MCP-powered workflows
- MCP platform operating at ≥ 99% availability, validated under peak load

Required Roles

- Platform engineering team (2–5 FTEs depending on scale)
- Security engineering (policy automation, anomaly detection on customer-data access)
- SRE / observability engineer
- AI enablement lead (user training & workflow design)
- Governance committee (security, platform, data protection, and business stakeholders)

Stage 4: MCP-Native Retailer

At Stage 4, MCP becomes part of the retailer's core fabric. AI assistants orchestrate complex workflows across commerce, supply chain, and customer systems, using governed context to automate tasks that once required human coordination. Business teams build and adapt their own assistants, while security and platform controls ensure safe, compliant execution. MCP is no longer perceived as AI tooling; it is embedded infrastructure enabling continuous innovation, faster decisions, and competitive advantage.

Organizational Symptoms

- AI-driven workflows are embedded in daily operations across stores, e-commerce, supply chain, and support
- Multi-MCP assistants automate recurring retail tasks with minimal human oversight
- MCP usage is no longer considered "AI", it is standard organizational plumbing
- AI governance and platform operations run as a mature, continuous program

Technical Markers

- Autonomous orchestration across MCP servers (agent-led or platform-led)
- Hybrid registries combining public, private, and SaaS MCP servers (Salesforce, SAP, Shopify, and internal)
- Dynamic permissioning and token exchange with least privilege, scoped per customer and per task
- Cost, performance, and context optimization tooling integrated
- Behavioral guards and policy engines evaluate AI actions before execution, which is critical when agents touch pricing and customer data
- Multi-model routing for agents across LLM providers

Risks if You Stay in This Stage (plateau risk)

- Innovation slows if the platform is not continuously updated as the MCP ecosystem evolves
- Governance drifts if AI oversight is not maintained as a continuous program
- Competitors who reached MCP-native status compound their advantage in personalization and operations
- Technical debt accumulates if agent and tool sprawl is not actively curated

Success Metrics (steady-state indicators)

- Majority of mission-critical retail workflows rely on MCP orchestration
- Banners and business units deploy or modify their own assistants with guardrails
- ≥ 95% audit log coverage across all MCP interactions
- Time to onboard a new team with MCP-enabled workflows: < 1 week
- Demonstrable revenue, margin, or velocity gains attributable to MCP (conversion, deflection, fulfillment efficiency)

Required Roles

- Platform & SRE team operating MCP as core infrastructure
- AI governance board ensuring responsible use of customer data
- Business workflow architects building and sharing assistant templates across banners
- Security analysts monitoring AI-enabled interactions and policies

Enterprise MCP Platforms

By now it is apparent that to advance through the stages of the Maturity Model, your retail organization will need an MCP platform. Today, an MCP platform must offer at least four core components:

1 Registry

The foundation of a platform is a catalog of trusted MCP servers, including official upstream servers, public vetted servers, and internally developed servers connecting to your commerce stack. All servers should eventually be stored and managed centrally for full control and visibility. The better registries let administrators permission and pre-configure every MCP server.

2 Runtime

Retailers must insist on running MCP servers in their private cloud; customer and commercial data must stay in your environment. A hardened runtime lets you containerize and deploy MCP servers to a Kubernetes cluster, using Kubernetes to orchestrate access policies, network endpoints, and more.

3 Gateway

The gateway ensures the integrity of your MCP estate. Most gateways manage security with a basic implementation of MCP-spec'd OAuth. Enterprise-worthy solutions push further and include a federated token exchange, full IdP integration, OTEL-driven analytics, and more. Every user and AI agent has a single, secure endpoint to access the context they need. In retail, this is also where you contain the blast radius of any single integration touching customer or payment systems.

4 Portal

For enterprise-wide adoption, it has to be dead simple for end users (especially non-technical knowledge workers) to discover the context they need and install MCP servers with a single click. An intuitive portal is a key element of any MCP platform.

Now that we have defined a platform, let's consider the decision to build vs. buy, and, if buying, the criteria to weigh.

Build or Buy an MCP Platform

If you have decided you need an MCP platform, the next decision is whether to build your own (proprietary or assembled point solutions) or buy a complete platform. Here are the criteria to consider:

Time-to-value	If you face limited internal pressure to stop shadow AI and centralize control (a 6+ month horizon), you can consider building. But in retail, where shadow AI is touching customer and pricing data and peak season is always approaching, the pressure to deploy a governed solution in weeks usually makes buying a proven solution the obvious choice.
Competency	If your engineering organization has spare capacity and genuine platform-building sensibilities, building is an option. If your teams are competing for resources or skew more towards commerce apps and customer-facing services than infrastructure platforms, you should likely buy.
Ongoing capacity	It is early days for MCP; both the protocol and the ecosystem are evolving fast. Even with resources at the outset, you will need ongoing capacity and an appetite for constant change to maintain a homegrown platform. If that is unlikely, a self-built platform will struggle with breaking integrations and outdated practices, and you will end up buying anyway.
Operational complexity	If you expect a small number of MCP servers for a defined set of departments, building is feasible. If you anticipate pursuing MCP-native status (deploying hundreds of servers across business units, automating lifecycle management, and meeting peak-season availability) it is time to buy a leading solution.

Criteria for Buying an MCP Platform

Most retailers will choose to buy, given the pressure to minimize time-to-value and maximize control. The MCP market is noisy and many platforms sound the same. Here are the criteria we have seen matter in practice:

Track record

Many entrants in the MCP ecosystem are vibe-coded gateways from first-time founders. A few providers have solutions running in production with large enterprises, led by people who have long served demanding customers. They understand enterprise requirements and complexity.

Completeness

An MCP platform includes four core components: registry, runtime, gateway and end-user portal. A provider offering only one or two will force integration work or proprietary effort onto your team, extending time-to-value. At present, only a small number of providers offer all four components at an enterprise standard.

Security

The MCP ecosystem faces both internal (shadow AI) and external (malicious actor) threats. For a retailer connecting agents to customer PII, payment data, and supplier systems, a strong security posture is the most critical capability of all. Minimum viable offerings address authentication via OAuth. Insist on more: federated token exchange, SSO and IdP integration, OTEL analytics, secrets management, and cryptographic server provenance checks.

Openness

Ensure any solution you buy has a sustainable future. Many MCP providers are startups; you can weigh investor strength and runway, but more confidence comes from an established open source core. An underlying open source project offers access to innovation and a path forward. Look specifically for projects with external maintainers as an indicator of momentum, relevance, and longevity.

Stacklok Enterprise MCP Platform

Stacklok offers a complete Enterprise MCP Platform that simplifies the deployment and management of MCP servers. Fortune 500 enterprises are using this platform in production to move through the Maturity Model and deliver more return on their AI investments.

The Stacklok Enterprise MCP Platform includes all four core components referenced above. These components are tightly integrated, and integrations extend to your existing authentication, observability, secrets management, and other solutions.

Deployment Modes

Stacklok's platform is available in three modes to suit different maturity stages:

UI

Desktop application for individual developers and analysts to discover, deploy, and manage MCP servers locally with a user-friendly interface.

CLI

Command line interface for quick deployment with advanced features like custom permissions and telemetry.

Kubernetes Operator

Enterprise-grade operator for teams to run MCP servers in multi-user, multi-banner environments with centralized management and security controls.

Key Enterprise Features

Secure by Default

Every server runs in isolated containers with only necessary permissions. Secrets managed securely, never in plaintext, which is critical when agents reach customer and payment systems.

Declarative Management

MCP Server Custom Resource Definitions enable GitOps workflows and infrastructure-as-code practices.

Observability

OpenTelemetry and Prometheus metrics bridge the monitoring gap in MCP deployments, with usage attributable by team and banner.

Enterprise Authentication

OAuth/OIDC SSO integration, secure token exchange, and audit logging that meets PCI DSS, GDPR, and CCPA requirements.

Multi-Namespace Support

Cluster-wide or namespace-scoped deployment modes following the principle of least privilege.

Kubernetes-native

Kubernetes operator allows you to use familiar patterns and declarative practices as you scale your MCP footprint.

For more information about the Stacklok Enterprise MCP Platform, visit stacklok.com, email **enterprise@stacklok.com**, or engage us via Discord at **https://discord.gg/stacklok**.

To try the ToolHive open source project, check out our GitHub repo: **https://github.com/stacklok/toolhive.**