

Abstract

The proliferation of AI agents capable of interacting with the real world necessitates a standardized approach to connecting them with external software services, such as APIs and databases. The current landscape is characterized by fragmented integrations, service-specific authentication complexities, and a lack of a universal mechanism for agents to discover and utilize capabilities programmatically.

This thesis presents the Universal MCP Python SDK, a framework designed to abstract these challenges by implementing and leveraging the Model-Context Protocol (MCP) standard. The SDK provides a robust layer for defining service capabilities as “tools” (callable functions), automatically generating structured metadata for agent consumption, and managing diverse authentication flows and transport mechanisms.

Key components include a flexible tool definition and management system, an application abstraction layer for building standardized service integrations, and a modular architecture for handling authentication and transport. The SDK integrates with a backend platform that serves as a central registry for applications and a provider for credential management, particularly through an AgentR-specific integration type.

This work details the design and implementation of the Universal MCP SDK, demonstrating how it enables developers to build reusable service integrations “Applications” that can be easily exposed via MCP servers and consumed by various AI agent frameworks. The resulting ecosystem promotes scalability, simplifies integration development, and accelerates the deployment of context-aware AI agents.

Keywords: Universal MCP, Model-Context Protocol, AI Agent, Tools, Service Integration, Python SDK, Authentication, API Integration, Tool Management, Pydantic.