

Morgan Stanley



MCP Servers for Your mainframe

Calder Sacks, Executive Director

August 2026

Bio Page

CALDER SACKS

Executive Director

Enterprise Z

Phone: +1 212 276-1952

Calder.Sacks@morganstanley.com

Years of Experience: 34 Years

Years at Morgan Stanley: 27 Years

Biography

Worked in many roles within the mainframe technology landscape. Started as a Natural mainframe programmer and moved into the Adabas DBA role. Lead the database engineering team at Morgan Stanley for 10+ years and now lead the team working on Mainframe integration strategy and technologies for our Cobol/DB2 mainframe environment.

Been working on technologies between mainframe and distributed since 2003.

- One of the first companies to call Java RPC services from our mainframe Batch and CICS processes back in 2005
- Setup Sybase access from the mainframe.
- Implemented SQL & ACI access to mainframe Adabas and CICS
- Running Kafka Connectors running on the mainframe to distributed Kafka

Our goal is integrating mainframe to be just another platform and to leverage what has already been built where possible

Table of Contents

Why MCP servers and API's for the mainframe ?

Types of Servers and API's

MCP server architecture and data flow

MCP for mainframe engineering and operations

Use cases: the mainframe in everyday tools

Tools, vendors and the open Zowe project

Summary

Why MCP servers and API's for the mainframe ?

- Get to the data at the source of the data
- Leverage what has already been built without reinventing everything
- Creating a set of standard interfaces for developers to discover and use to build new workflows and processes is key to many organizations
- The ability to discover MCP servers functions, API's and micro services will become more important as Agentic AI is used to build new processes and workflows

Types of API's

- There are many API's that have been built over the years to bridge applications
- Many are custom built and the standards and interfaces agreed upon by a few development teams
- Direct connection to applications on mainframes have generally been purpose built

DB2 REST services

- Taken all of our DB2 Stored procedures and exposed them as REST services

CICS REST services

- Creating REST services for access to our CICS transactions

ZOWE and the mediation layer

- Leveraging ZOWE and the mediation layer to expose and use mainframe components

Vendor interfaces (Sysview etc)

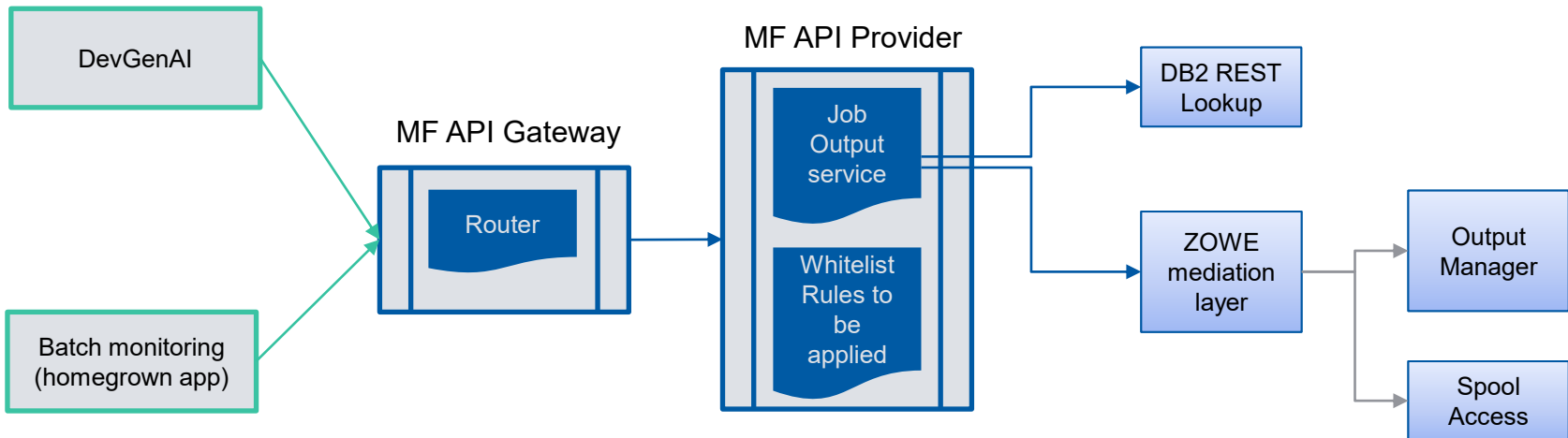
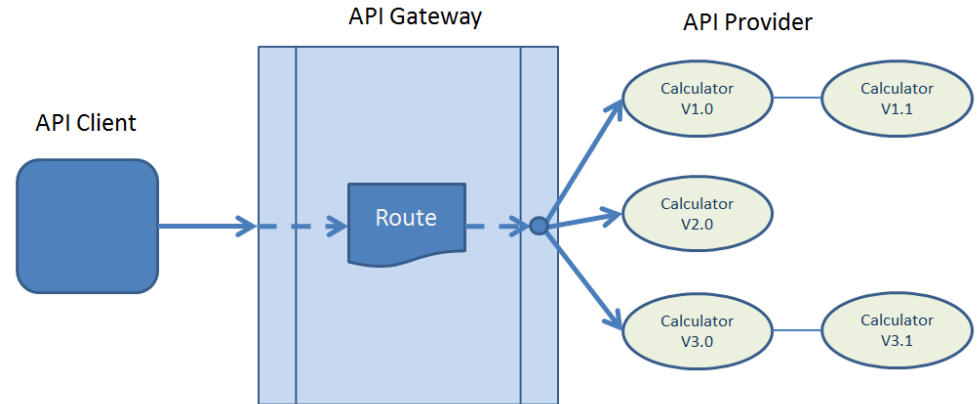
- Using API's provided by vendor products

Python to build services on the mainframe

- Building scripts and services to allow interaction with the mainframe

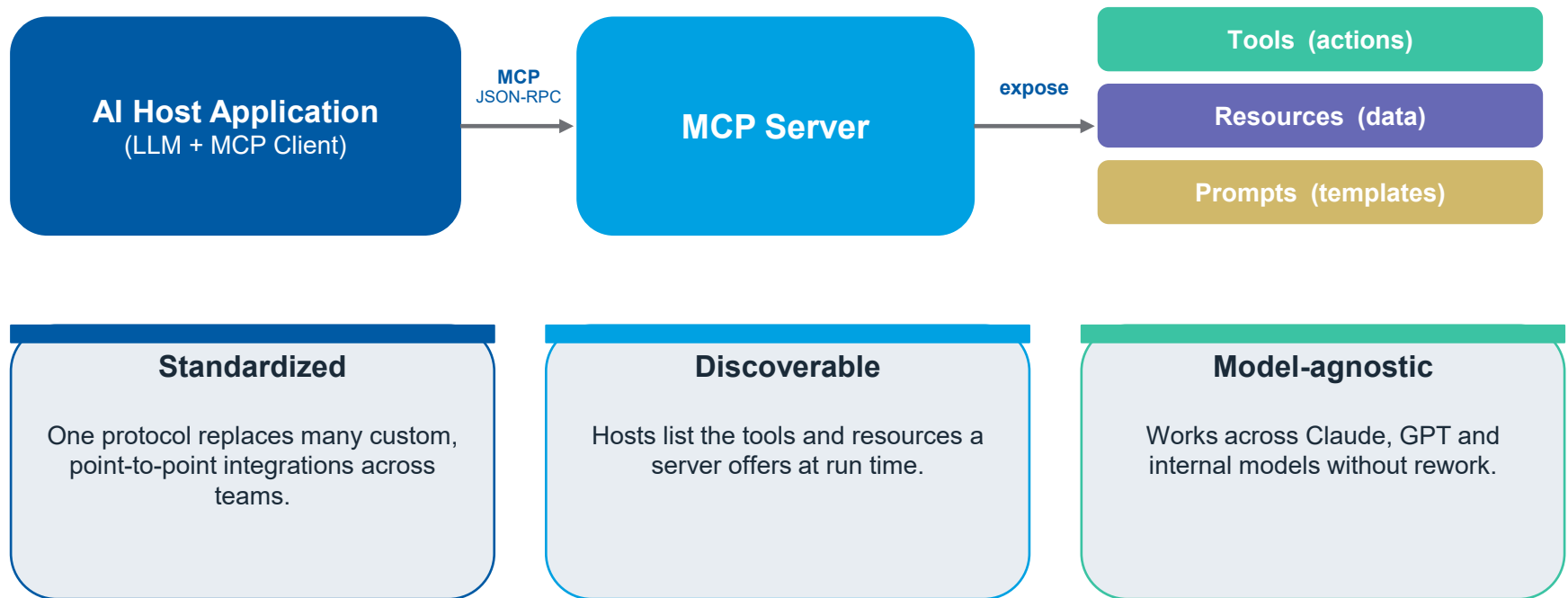
API Gateways and providers

- Gateways can be used in many contexts
 - Allowing access to different versions
 - Controlling volume flow
 - Simplifying connectivity
 - Ensuring security and compliance
 - Reducing complexity for consumers
 - Combining sources to give a single result



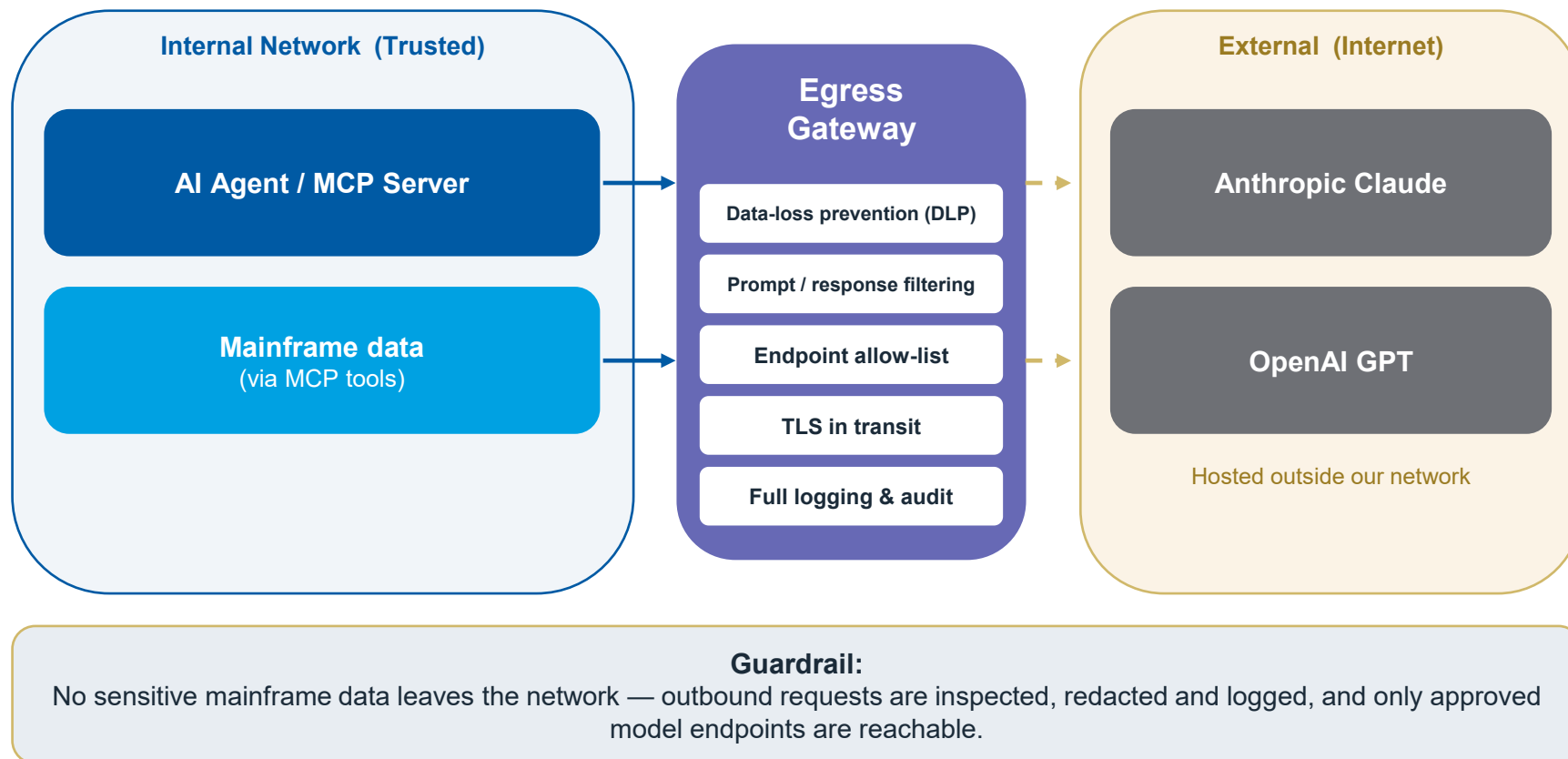
What is an MCP Server?

The Model Context Protocol (MCP) is an open standard that lets AI applications connect to tools, data, and systems through one consistent interface — so a model can discover and call capabilities without custom, point-to-point integrations.



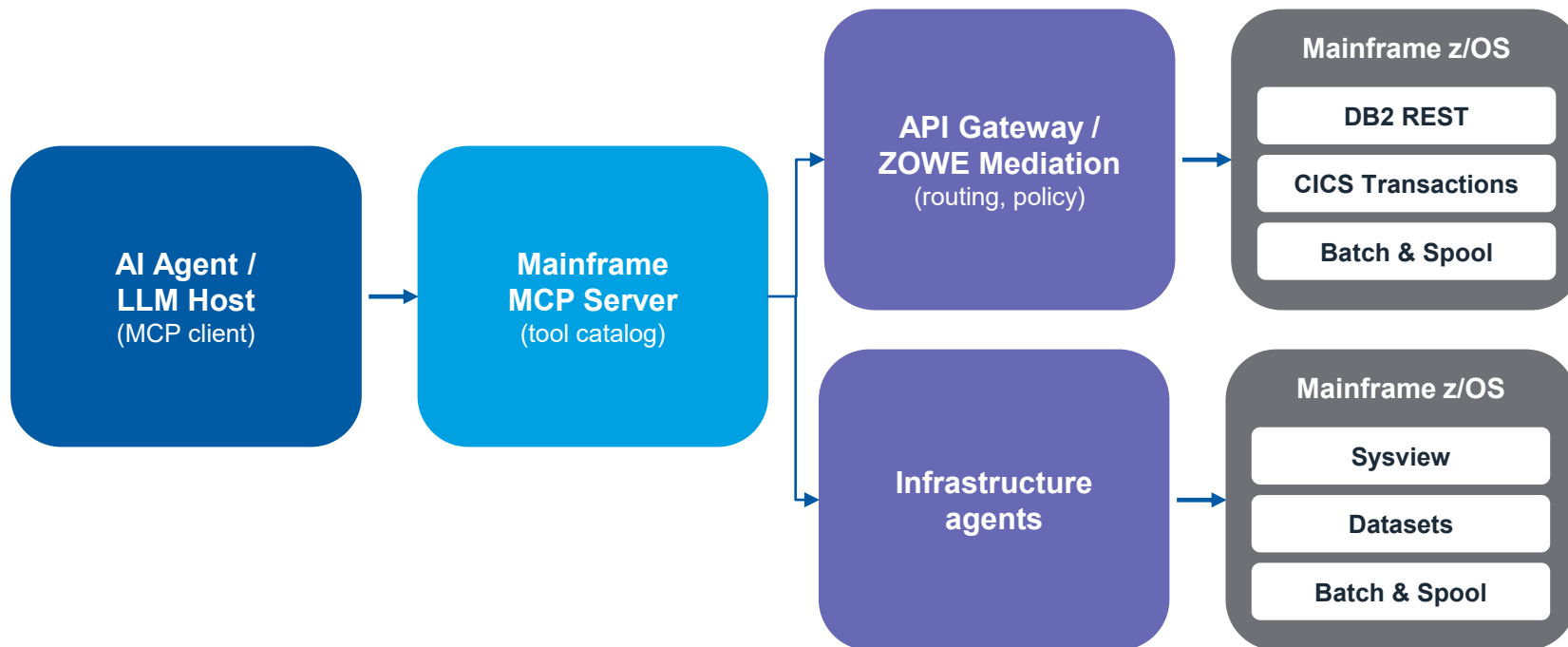
Connecting to External LLMs — Egress & Guardrails

LLMs hosted outside our network (Anthropic Claude, OpenAI GPT) are reached only through a controlled egress path — inspected, redacted and logged.



MCP Server Architecture for the Mainframe

The MCP server is the bridge — it presents mainframe capabilities to LLM's and reuses the REST services we have already built on z/OS.

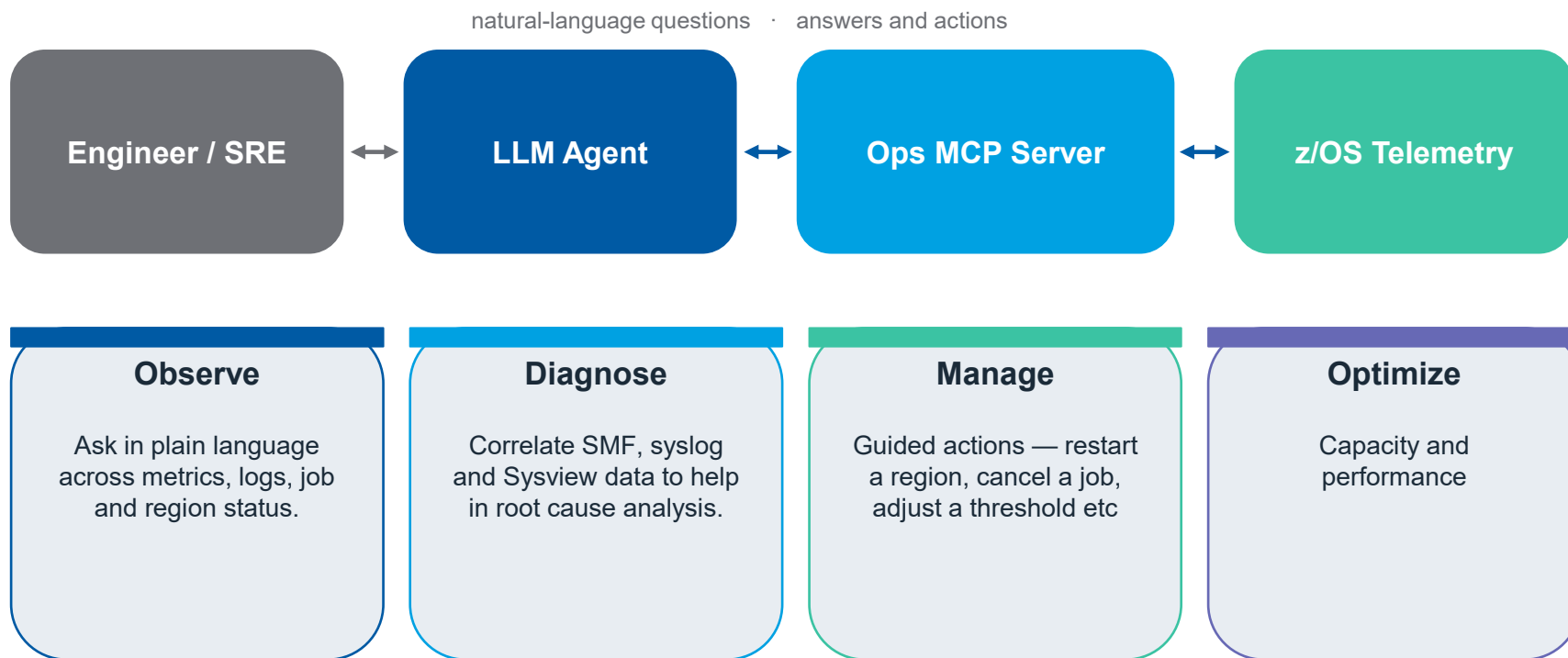


Reuse, don't rebuild:

The MCP server maps tool calls onto existing services through the gateway — adding discovery, scoping and audit without new point-to-point code.

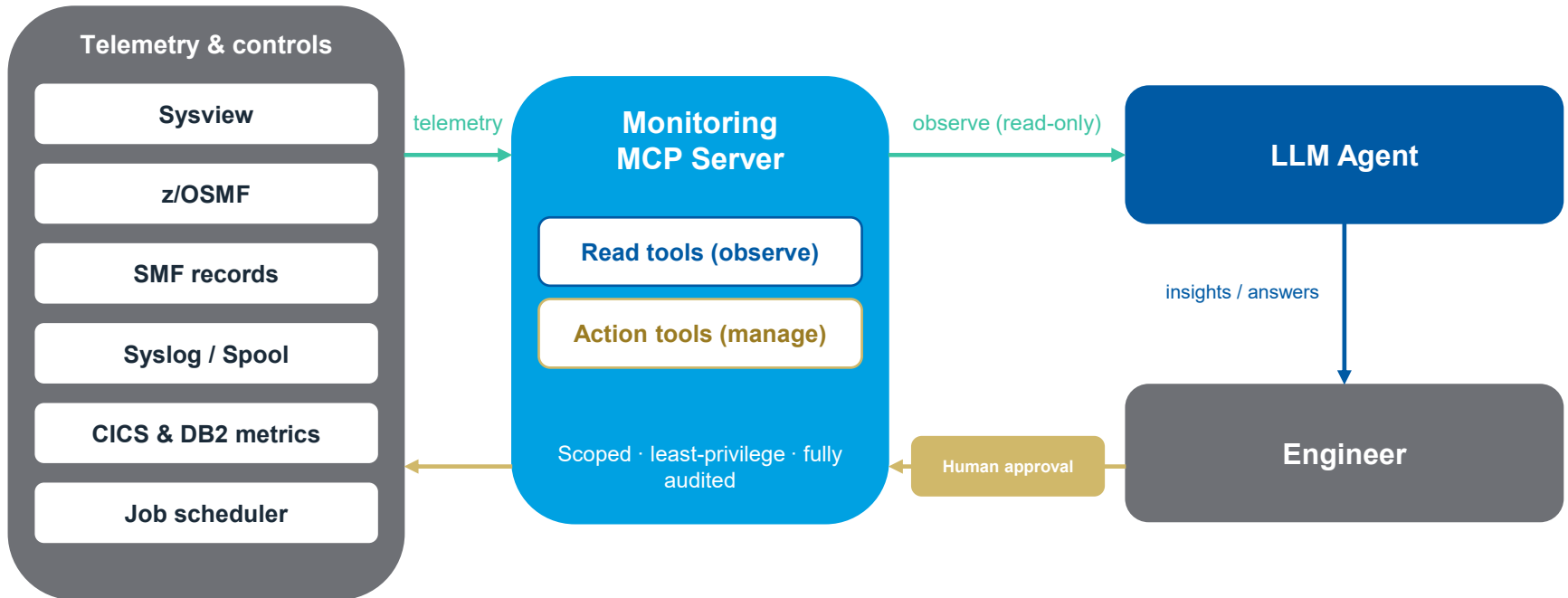
MCP for Mainframe Engineering & Operations

Beyond application data, MCP servers can expose the mainframe's own operational telemetry — giving engineers a natural-language interface to observe, diagnose and manage the z/OS environment.



AI-Assisted Monitoring & Management

One monitoring MCP server, two lanes: a read-only observe path and an approval-gated management path.

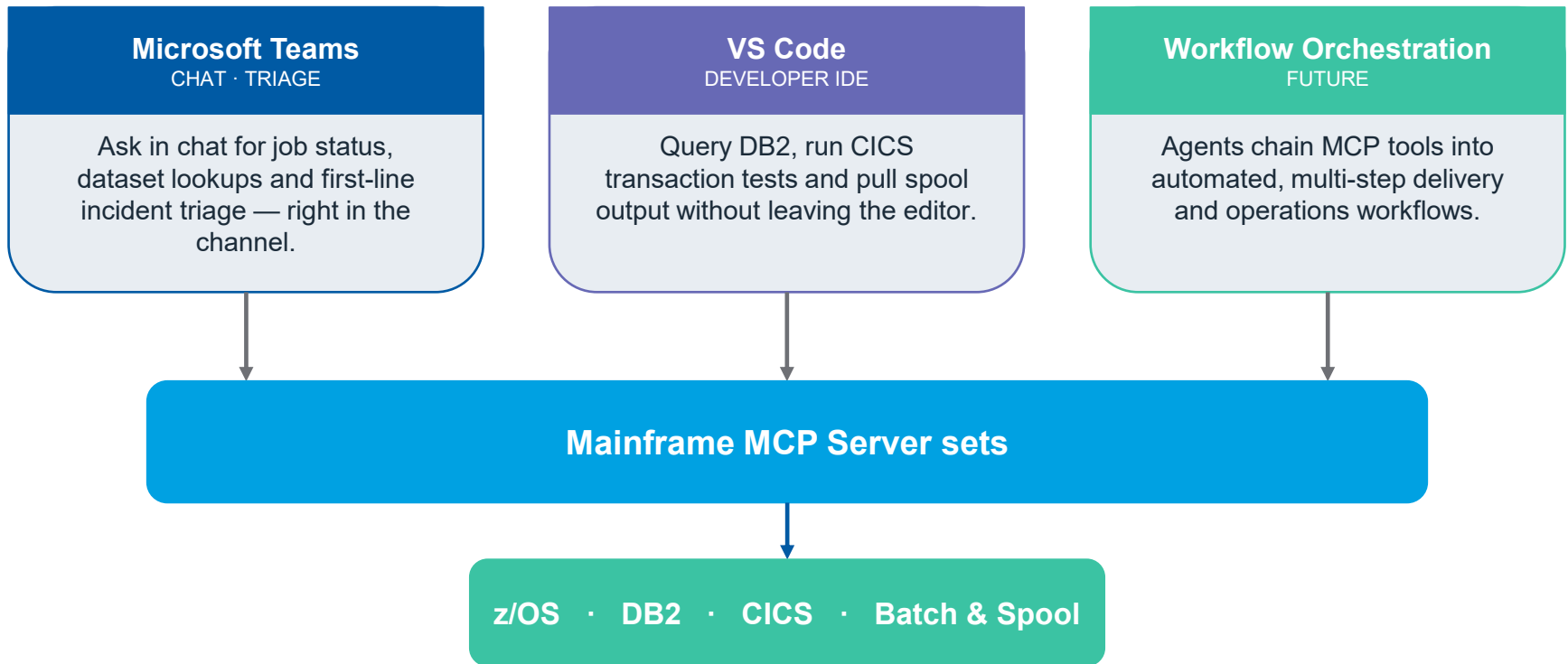


Guardrail:

Read-only by default. Management actions are scoped, require human approval and are fully logged — the LLM recommends, engineers decide.

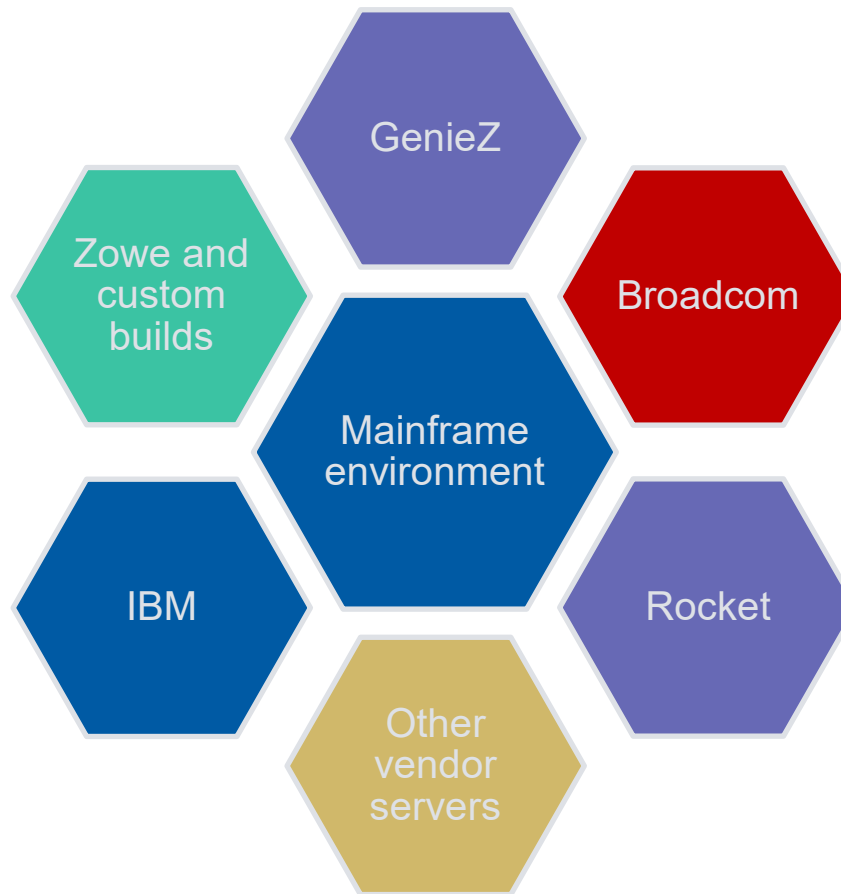
Use Cases: The Mainframe in Everyday Tools

The same MCP servers lets people and processes reach the mainframe from the tools they already using at the time



The Mainframe MCP Ecosystem

We looking at using a variety of MCP servers/services from all of our vendors that plan to build these interfaces as well any custom built servers we may need



Summary

- Extend the strategy with MCP servers so agentic AI and LLMs can safely discover and use mainframe capabilities
- Leverage microservices technology and REST services across products
- Plan to integrate with the larger Services catalog in order to make these API's and services discoverable
- Plan for mechanisms to manage this new eco system in terms of
 - Version control
 - Performance and connectivity
 - Access control
 - OIDC for connections
 - Least privileged model on the mainframe
 - Guardrails for any automated actions to prevent disaster

Questions ?