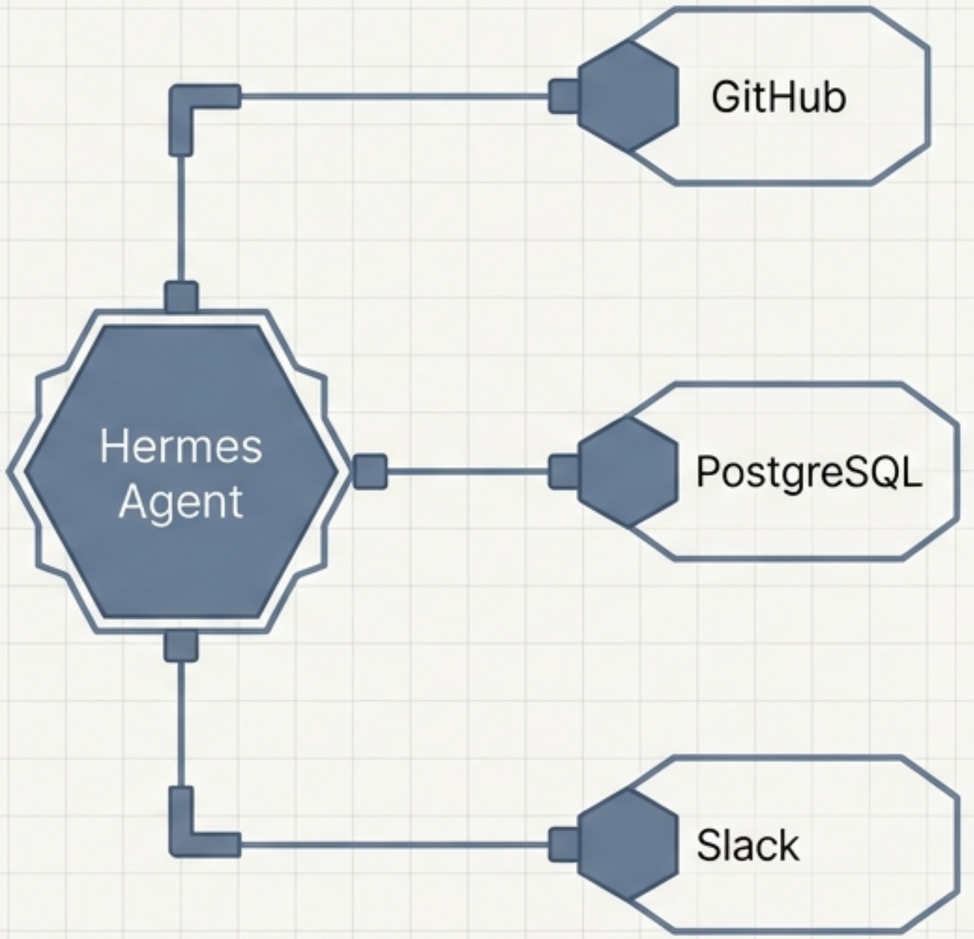
// BLUEPRINT DE SISTEMAS
// AUDIENCIA: INGENIERÍA / ARQUITECTURA

Arquitectura de Extensibilidad: Orquestación de Herramientas con MCP

Patrones de diseño para sistemas multi-agente, integración de Model Context Protocol y arquitecturas de aprendizaje continuo.

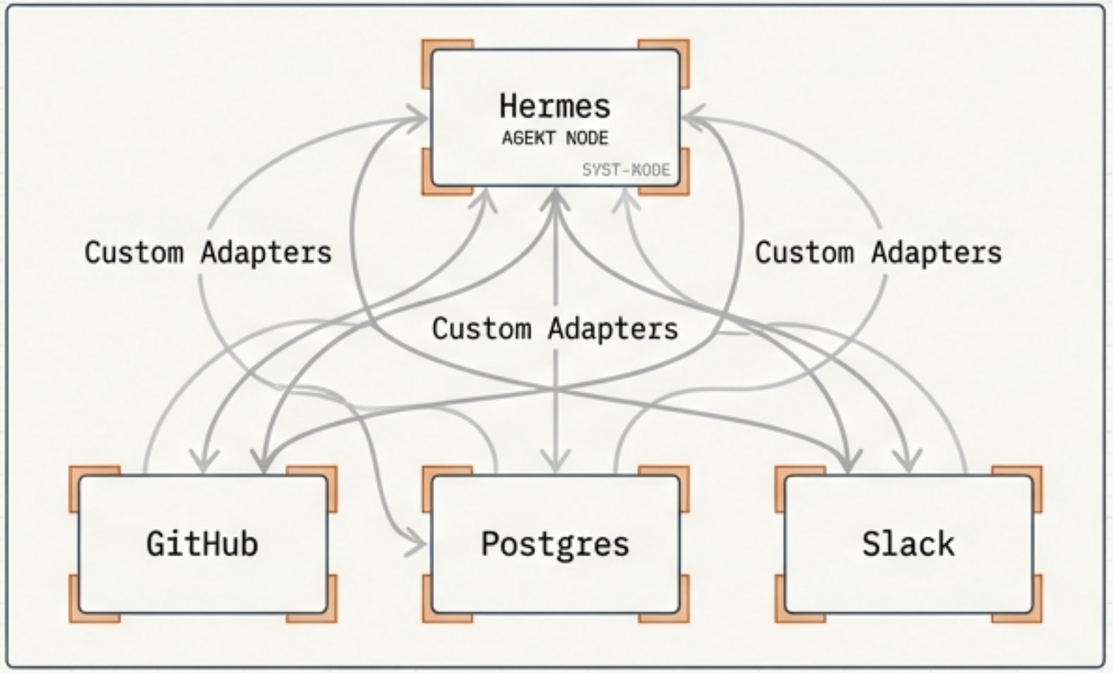


ESTÁNDAR: ANTHROPIC 2024 | ESTADO: DEPLOYMENT-READY

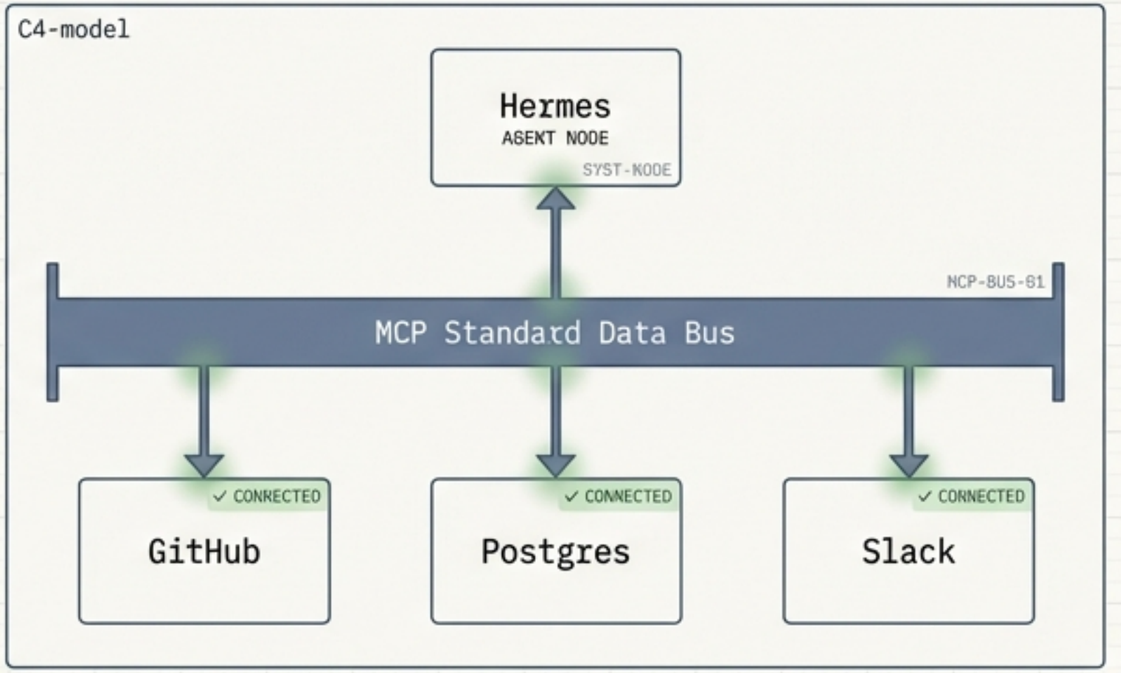
El Paradigma MCP: El Puerto USB del Ecosistema de IA

El Model Context Protocol (estándar abierto, 2024) elimina la necesidad de escribir adaptadores personalizados para cada servicio.

El Problema: Arquitectura Acoplada



La Solución: Bus de Datos Universal MCP

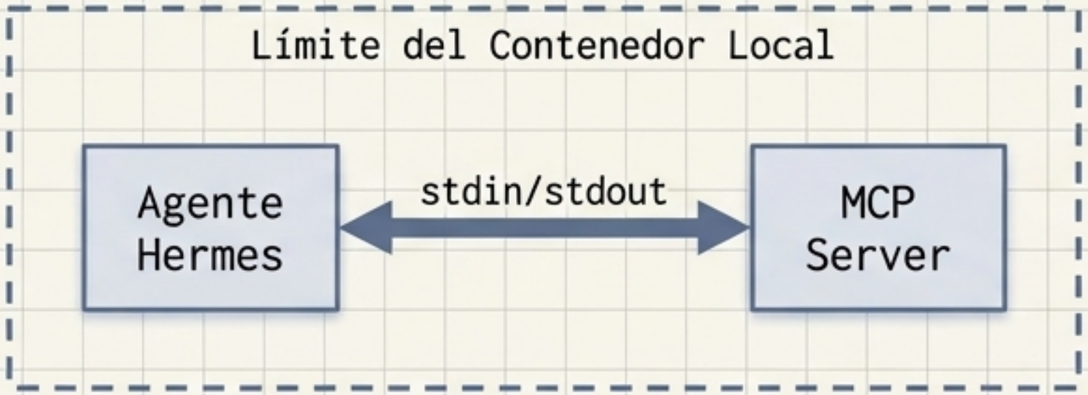


Línea Base & Escalabilidad

- **Línea Base:** Hermmes posee más de 40 built-in tools de alta capacidad.
- **Escalabilidad:** MCP permite conectar miles de servidores listos para usar sin código adicional.

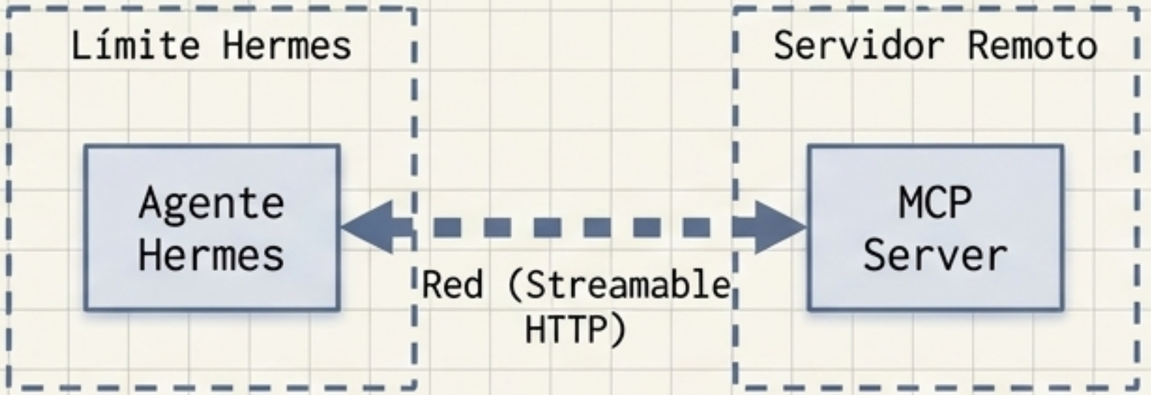
Topología de Despliegue: stdio vs. Streamable HTTP

stdio (Modo Local)



- **Ubicación:** Proceso hijo local
- **Ventaja:** Latencia mínima, zero network overhead
- **Caso Ideal:** Sistemas de archivos, bases de datos directas

HTTP (Modo Remoto)



- **Ubicación:** Servidor remoto independiente
- **Ventaja:** Despliegues aislados y compartidos
- **Caso Ideal:** Microservicios nube, uso por múltiples agentes

Veredicto Arquitectónico: stdio es suficiente para la mayoría de arquitecturas de agente único por su baja fricción y alto rendimiento.

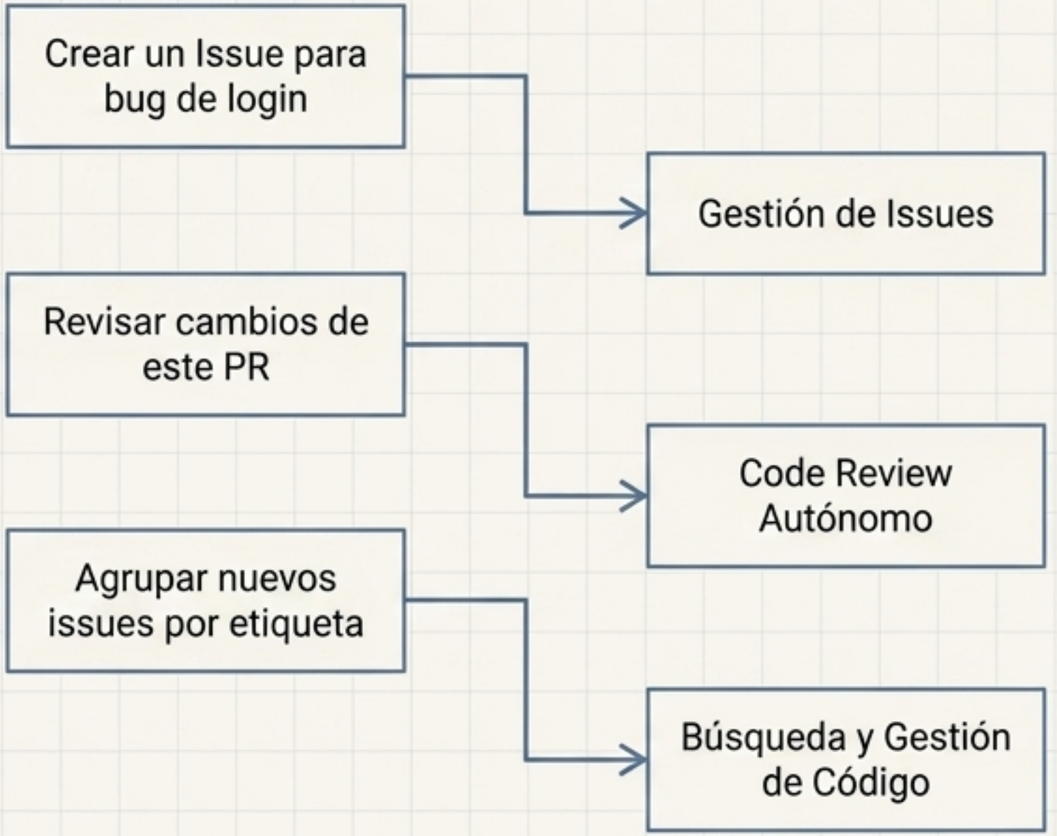
Implementación de Infraestructura: GitHub MCP

El Nodo de Configuración

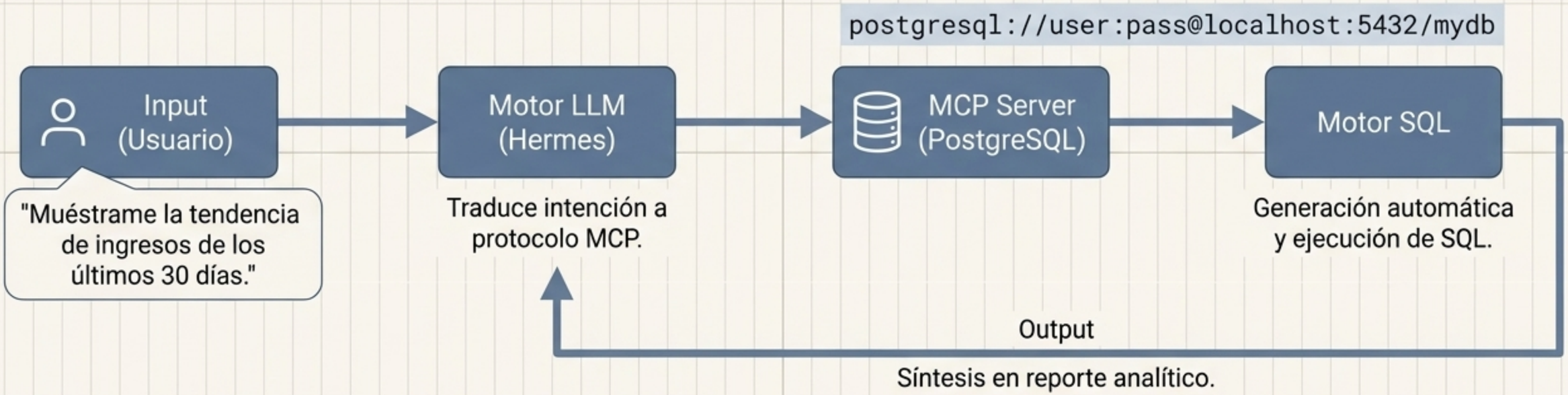
```
mcp_servers:
  github:
    command: "npx"
    args: ["-y", "@modelcontextprotocol/server-github"]
    env:
      GITHUB_PERSONAL_ACCESS_TOKEN: "${GITHUB_TOKEN}"
```

Inyección segura. Utilice variables de entorno en lugar de hardcoding. Requiere permisos repo, read:org, issues y pull_requests.

Capacidades Habilitadas



Data Pipelines: Integración SQL Autónoma (Text-to-SQL)

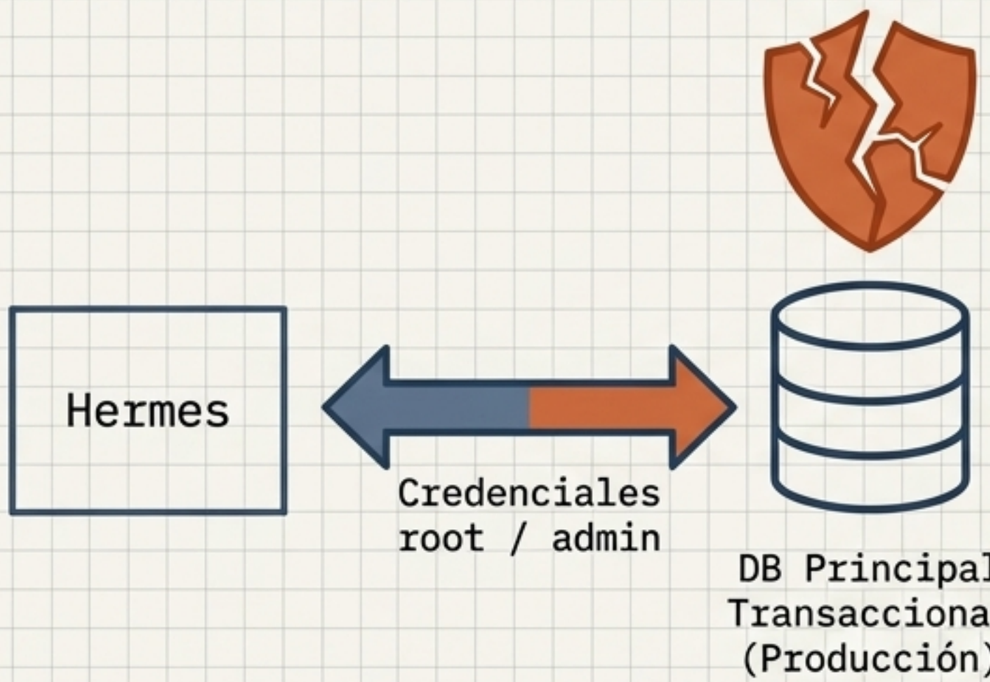


Soportado nativamente para PostgreSQL, MySQL y SQLite simplemente intercambiando el connection string y el paquete del servidor.

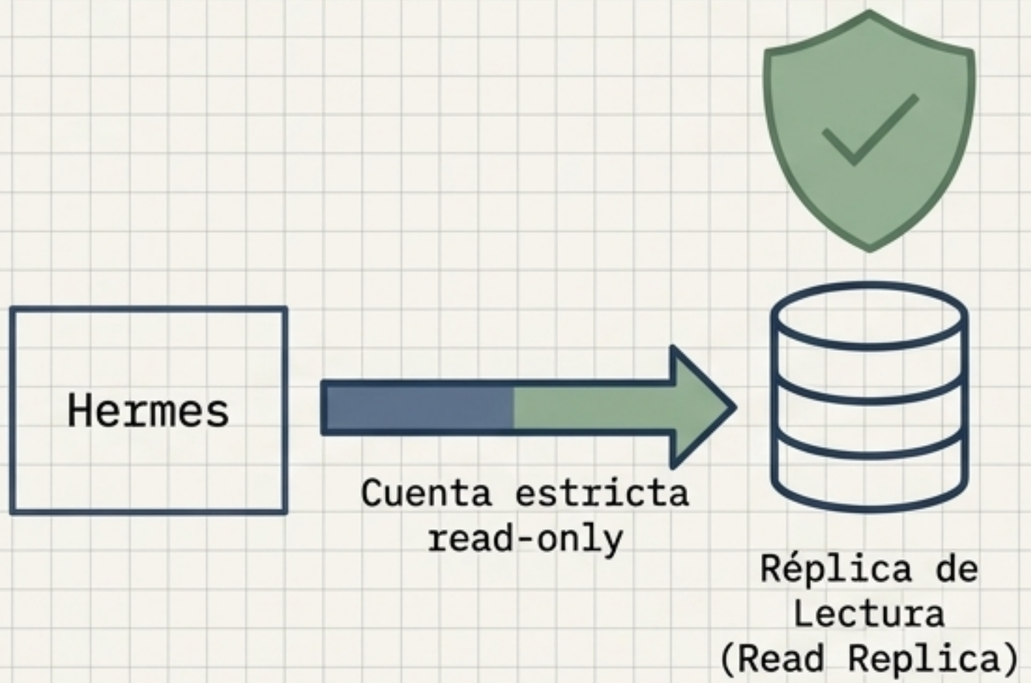
Postura de Seguridad: Arquitectura Zero-Trust y PoLP

CUIDADO: Los servidores MCP de bases de datos tienen acceso de LECTURA-ESCRITURA por defecto.
Un agente autónomo con permisos de escritura en producción representa un vector de riesgo inaceptable.

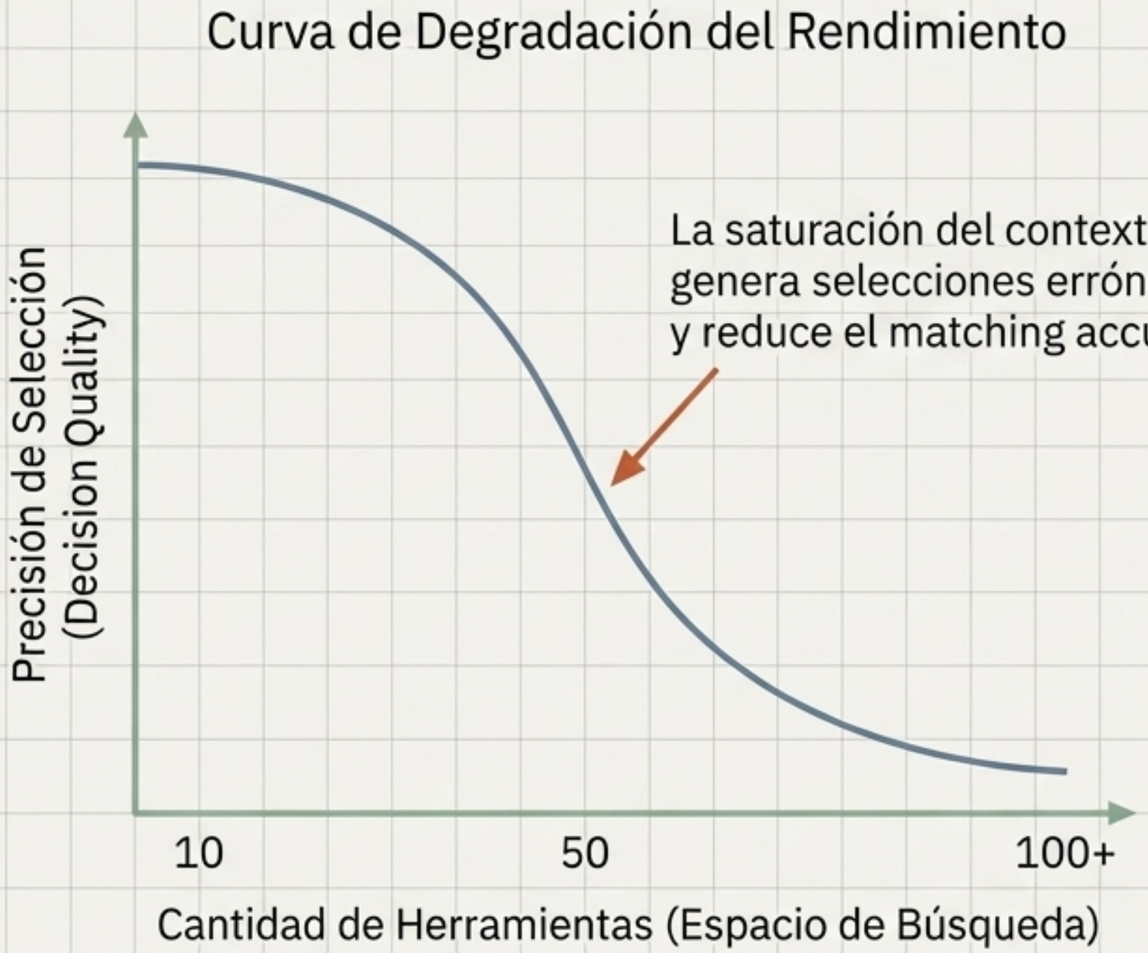
Anti-Patrón (Riesgo Crítico)



Patrón Recomendado (Seguro)



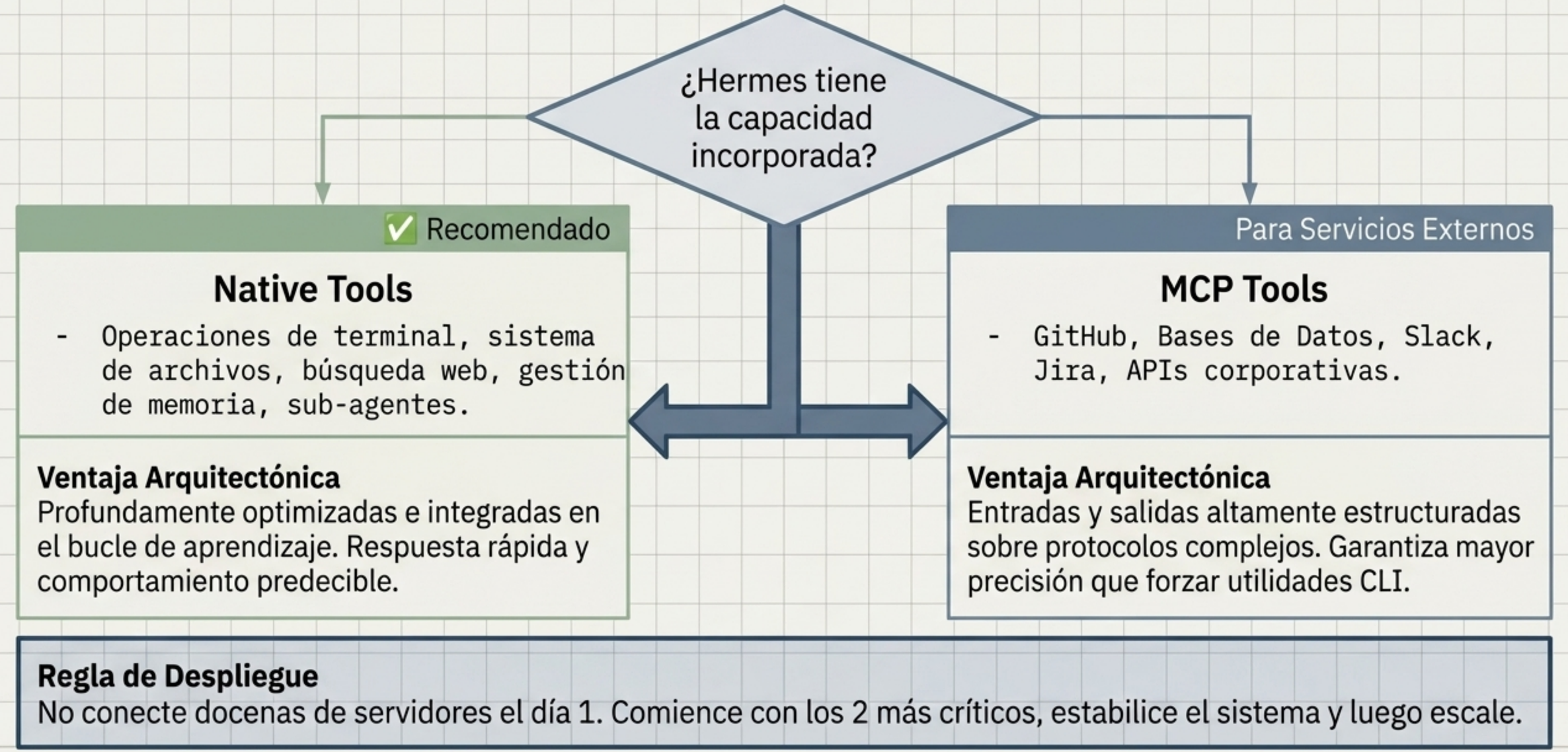
Optimización Cognitiva: El Cuello de Botella de Decisión



Solución: Per-Server Tool Filtering.
El principio de menor privilegio importa más en la era de los Agentes que nunca antes.

```
allowed_tools:
  - "list_issues"
  - "create_issue"
  - "get_pull_request"
```

Matriz de Decisión: Native Tools vs. MCP Tools



Regla de Despliegue
No conecte docenas de servidores el día 1. Comience con los 2 más críticos, estabilice el sistema y luego escale.

La Tríada de Autonomía: MCP + Skills + Native Tools

Ingeniería	GitHub MCP (Lectura de Diffs) + Code Review Skill (Criterios de calidad) = Revisión de Código Autónoma en PRs
Operaciones	Postgres MCP (Acceso SQL) + Weekly Report Skill (Métricas clave) + Cron Job Nativo = Extracción y publicación en Slack

La autonomía real surge de orquestar protocolos estandarizados bajo lógicas de negocio definidas.