

Introducción a LLMs y Agentes

Arquitecturas y Aplicaciones en Ingeniería de Software

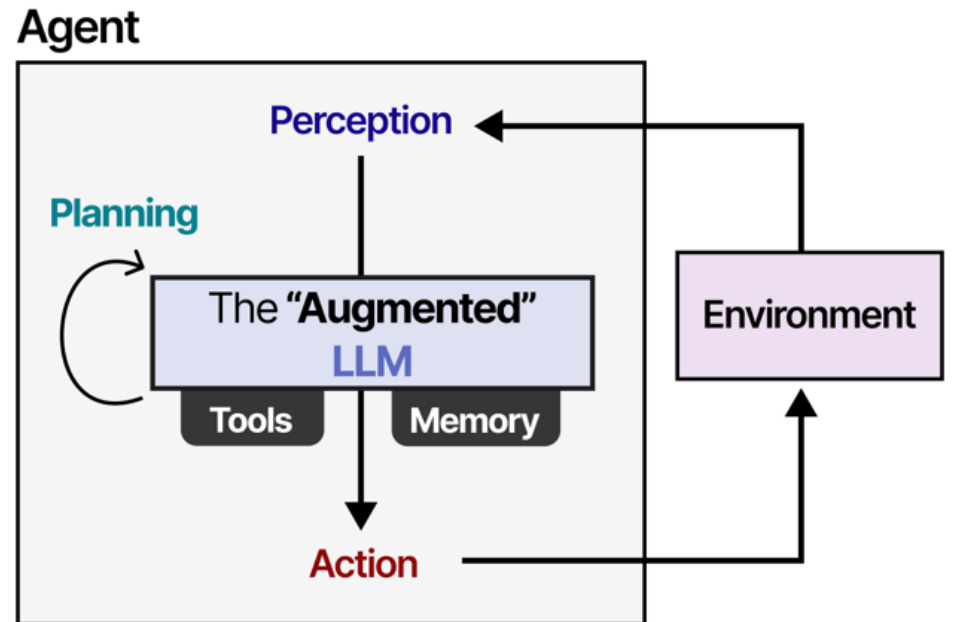
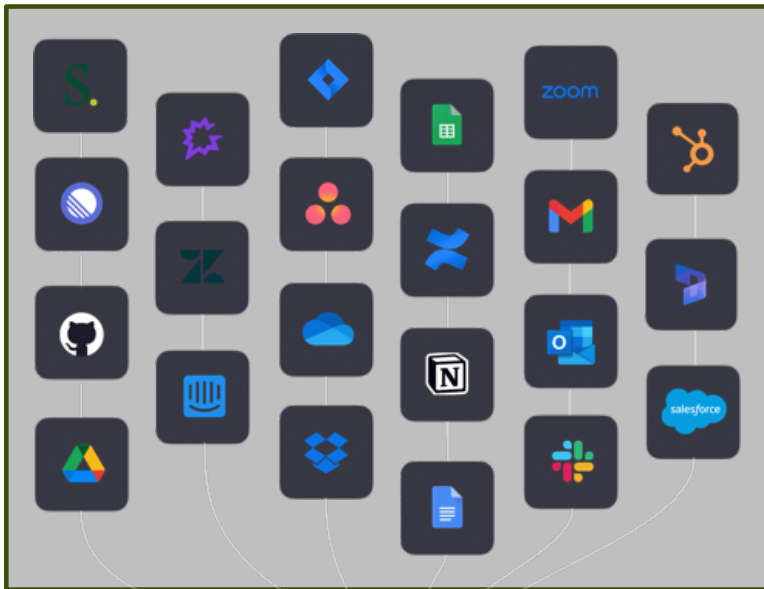
IV ESCUELA DE INFORMÁTICA

J. ANDRÉS DÍAZ PACE
2025

MCP **(Model Context Protocol)**

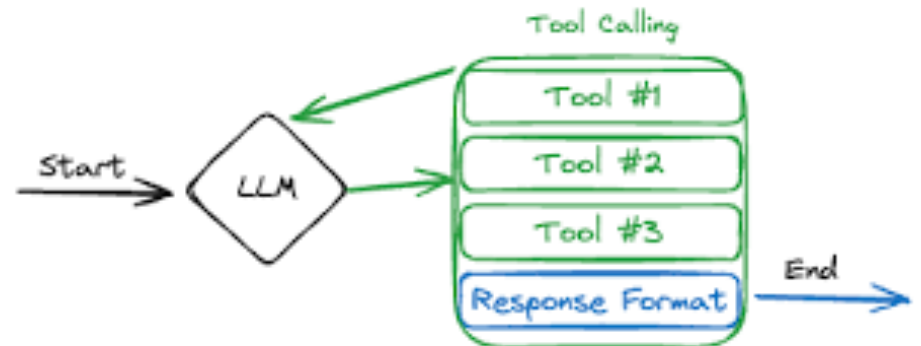


Agentes “todopoderosos” con tools



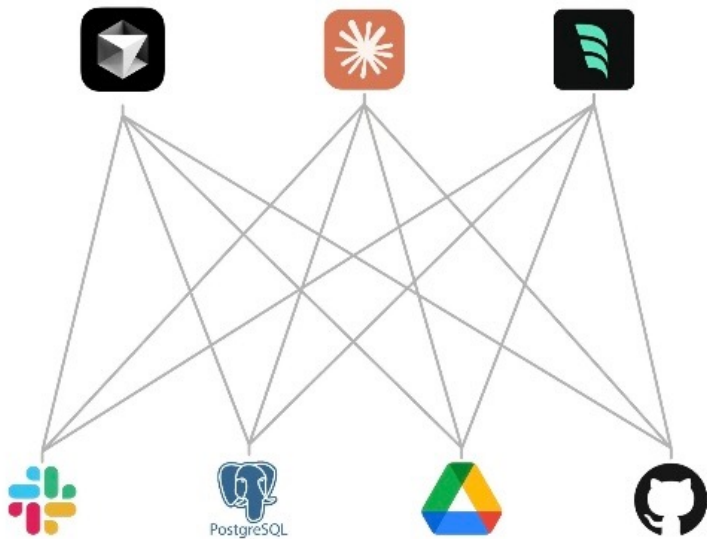
¿Que es MCP (Model Context Protocol)?

- Es un protocolo open-source, desarrollado por Anthropic, que facilita la **integración** entre:
 - aplicaciones basadas en IA / agentes / LLMs
 - tools y fuentes de datos

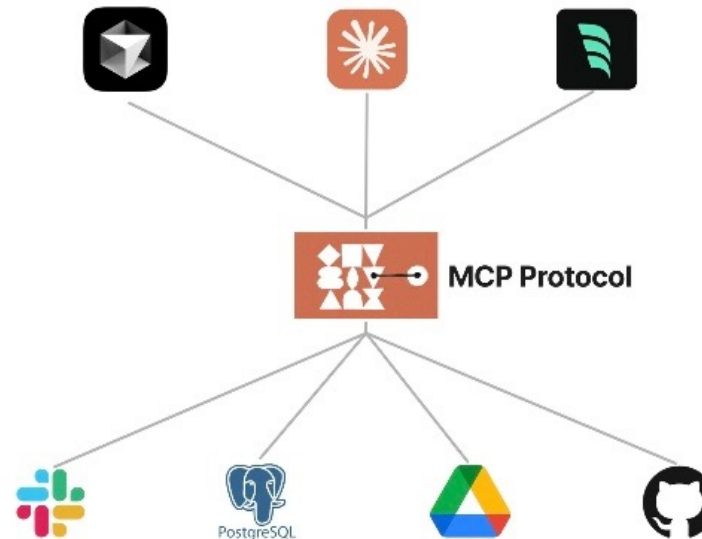


MCP como protocolo (estándar)

Without MCP

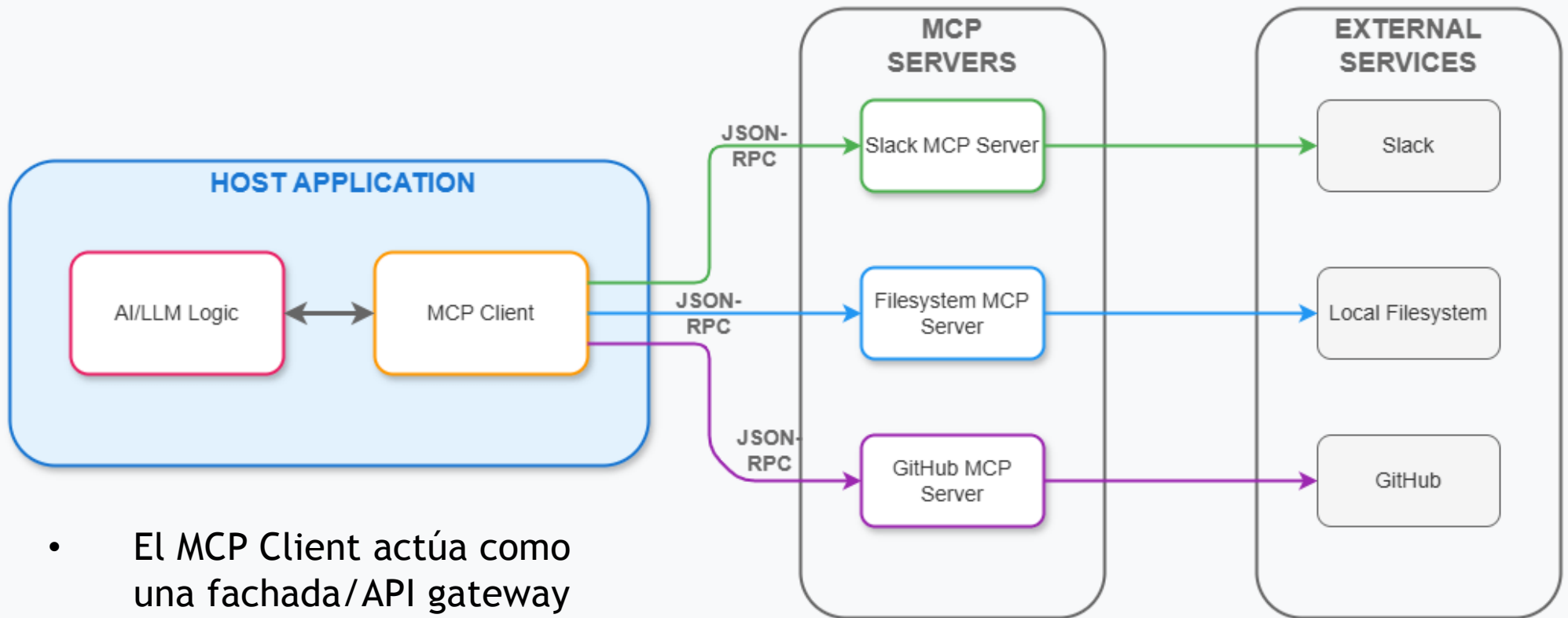


With MCP



Arquitectura MCP

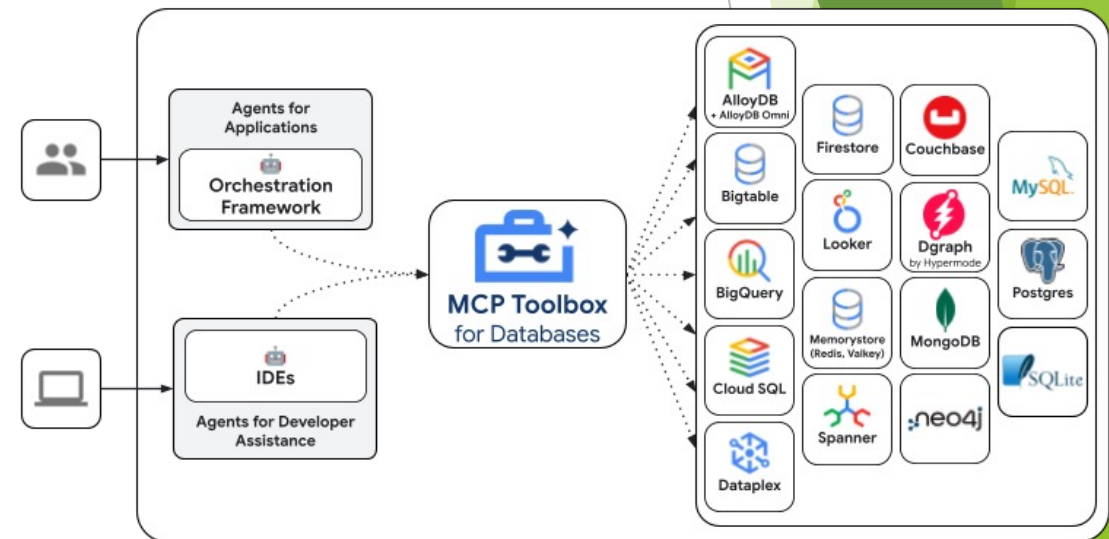
- El MCP Server actúa como un adapter



- El MCP Client actúa como una fachada/API gateway

No solo integración de tools ...

- Además de conectar tools (y sus resultados), MCP permite que el cliente se comuniquen con:
 - **Recursos**: fuentes de datos (read-only) tales como archivos, esquemas de BDs, documentación de APIs
 - **Prompts**
- Tipos de transporte
 - **Stdio** (local)
 - **HTTP con SSE** (remoto)
 - **Streamable HTTP** (remoto)
- Desarrollo de servers propios
- Comunicación con servers de terceras partes



FASTMCP

Ejemplo: FastMCP

```
from fastmcp import FastMCP

mcp = FastMCP("My MCP Server")

@mcp.tool
def greet(name: str) -> str:
    return f"Hello, {name}!"

# stdio
if __name__ == "__main__":
    mcp.run()

# http
if __name__ == "__main__":
    mcp.run(transport="http", port=8000)
```

```
import asyncio
from fastmcp import Client

client =
Client("http://localhost:8000/mcp")

async def call_tool(name: str):
    async with client:
        result = await
client.call_tool("greet", {"name":
name})
        print(result)

asyncio.run(call_tool("Ford"))
```

<https://github.com/jlowin/fastmcp>

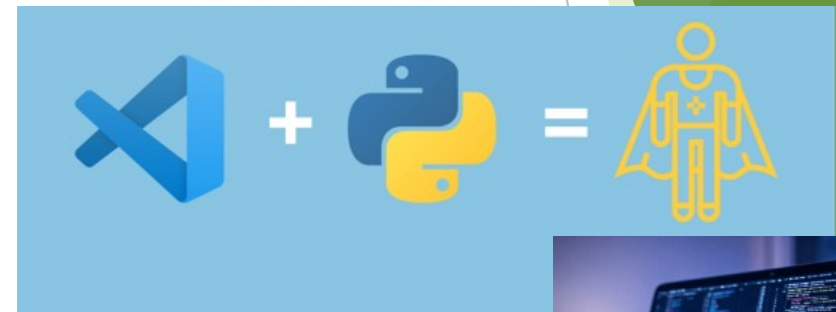
Challenge 1: ¿Un RAG sobre MCP?



FASTMCP

Pasos

- RAG básico sobre Langchain
 - Base vectorial para documentos
 - Ingesta (chunking)
 - Prompt
 - Cadena
 - Re-ranking de chunks (opcional)
- Definición de servidor MCP con FastMCP
- Definición de cliente
- Integración de cliente

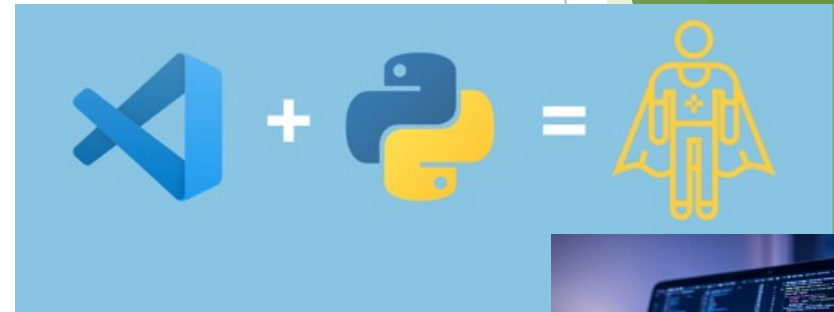


Challenge 2: ¿Un agente con tools MCP?



Pasos

- *Combinación de un acceso a issues y un chequeo contra las fuentes de documentación*



- Encapsulamiento de tools
- Definición de agente en Langchain



Desafíos

- Seguridad
- Gestión de gran cantidad de tools por un agente (filtrado, indexado, RAG, etc.)
- Buenas prácticas de descripción de tools, manejo de errores, etc.
- Validación de tools
- Modelado y **orquestación** de tools (no simplemente API calls)

Microsoft Research Blog

Tool-space interference in the MCP era: Designing for agent compatibility at scale

Published September 11, 2025

By [Adam Fourney](#), Senior Principal Researcher; [Tyler Payne](#), Senior Research Software Engineer; [Maya Murad](#), Senior Technical PM, AI Frontiers; [Saleema Amershi](#), Partner Research Manager

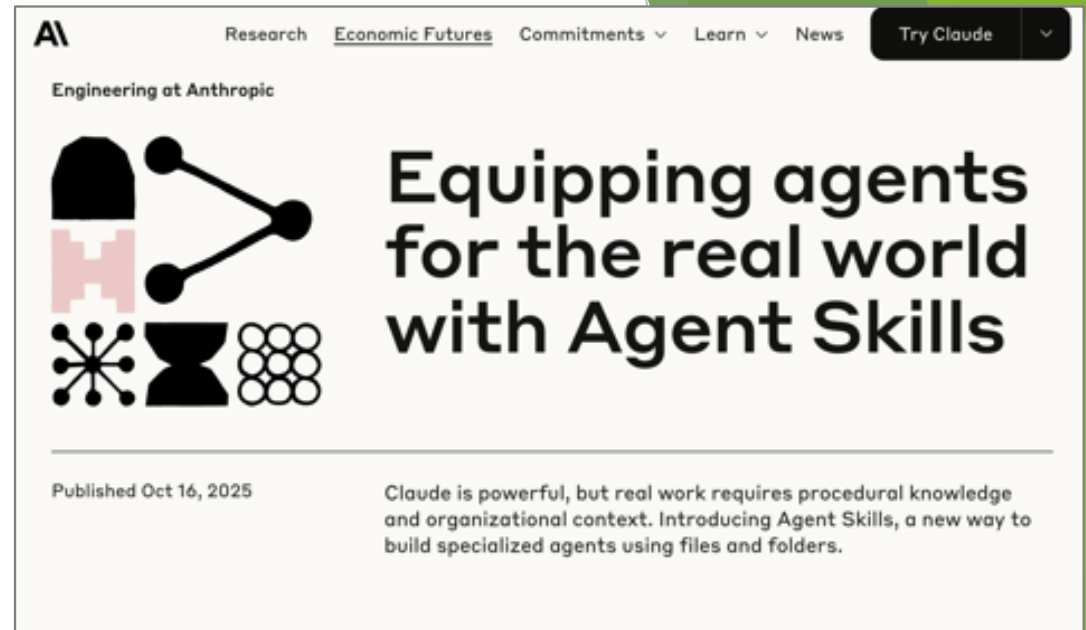
<https://www.microsoft.com/en-us/research/blog/tool-space-interference-in-the-mcp-era-designing-for-agent-compatibility-at-scale/>

Claude Skills

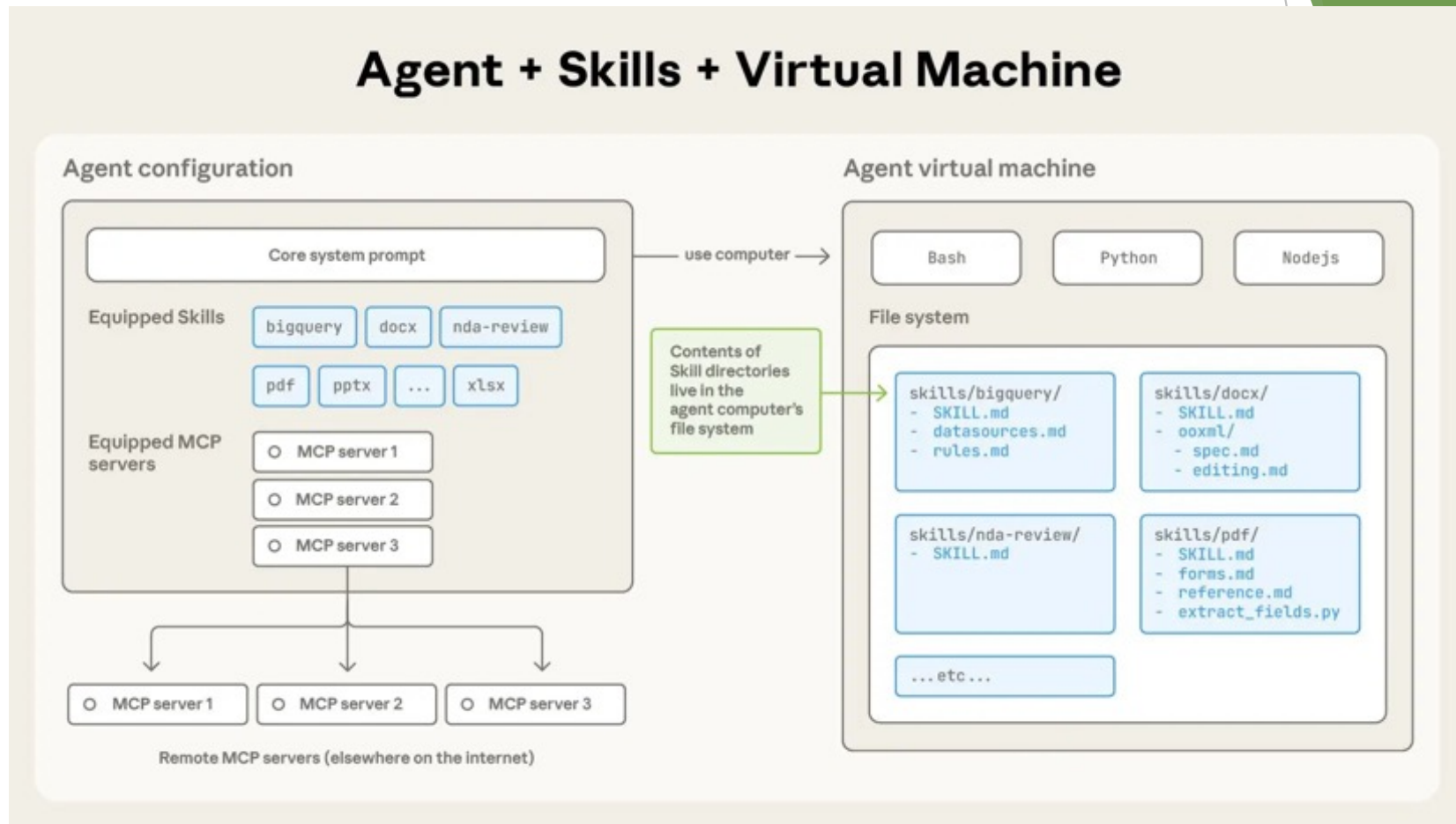
- Mecanismo mas “liviano” que MCP
- Expertise específica de dominio, mediante recursos modulares y que pueden cargar “mas contexto” solo cuando es necesario (progressive disclosure)
 - Ej., metáfora de TOC, capítulos, apéndices ...
- Aprovechan el mecanismo de acceso del agente a archivos (de instrucciones) en un file system

<https://www.anthropic.com/news/skills>

<https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills>



Claude Skills



Claude Skills

pdf/SKILL.md

YAML Frontmatter

```
name: pdf
description: Comprehensive PDF toolkit for extracting text and
tables, merging/splitting documents, and filling-out forms.
```

Markdown

Overview

This guide covers essential PDF processing operations using Python libraries and command-line tools. For advanced features, JavaScript libraries, and detailed examples, see `./reference.md`. If you need to fill out a PDF form, read `./forms.md` and follow its instructions.

Quick Start

```
...python
from pypdf import PdfReader, PdfWriter
```

```
# Read a PDF
reader = PdfReader("document.pdf")
print(f"Pages: {len(reader.pages)}")
```

```
# Extract text
text = ""
for page in reader.pages:
    text += page.extract_text()
```

pdf/reference.md

PDF Processing Advanced Reference

This document contains advanced PDF processing features, detailed examples, and additional libraries not covered in the main skill instructions.

pypdfium2 Library (Apache/BSD License)

Overview

pypdfium2 is a Python binding for PDFium (Chromium's PDF library). It's excellent for fast PDF rendering, image generation, and serves as ...

pdf/forms.md

If you need to fill out a PDF form, first check to see if the PDF has fillable form fields. Run this script from this file's directory:

```
`python scripts/check_fillable_fields <file.pdf>`,
and depending on the result go to either the "Fillable
fields" or "Non-fillable fields" and follow those
instructions.
```

Fillable fields

If the PDF has fillable form fields:

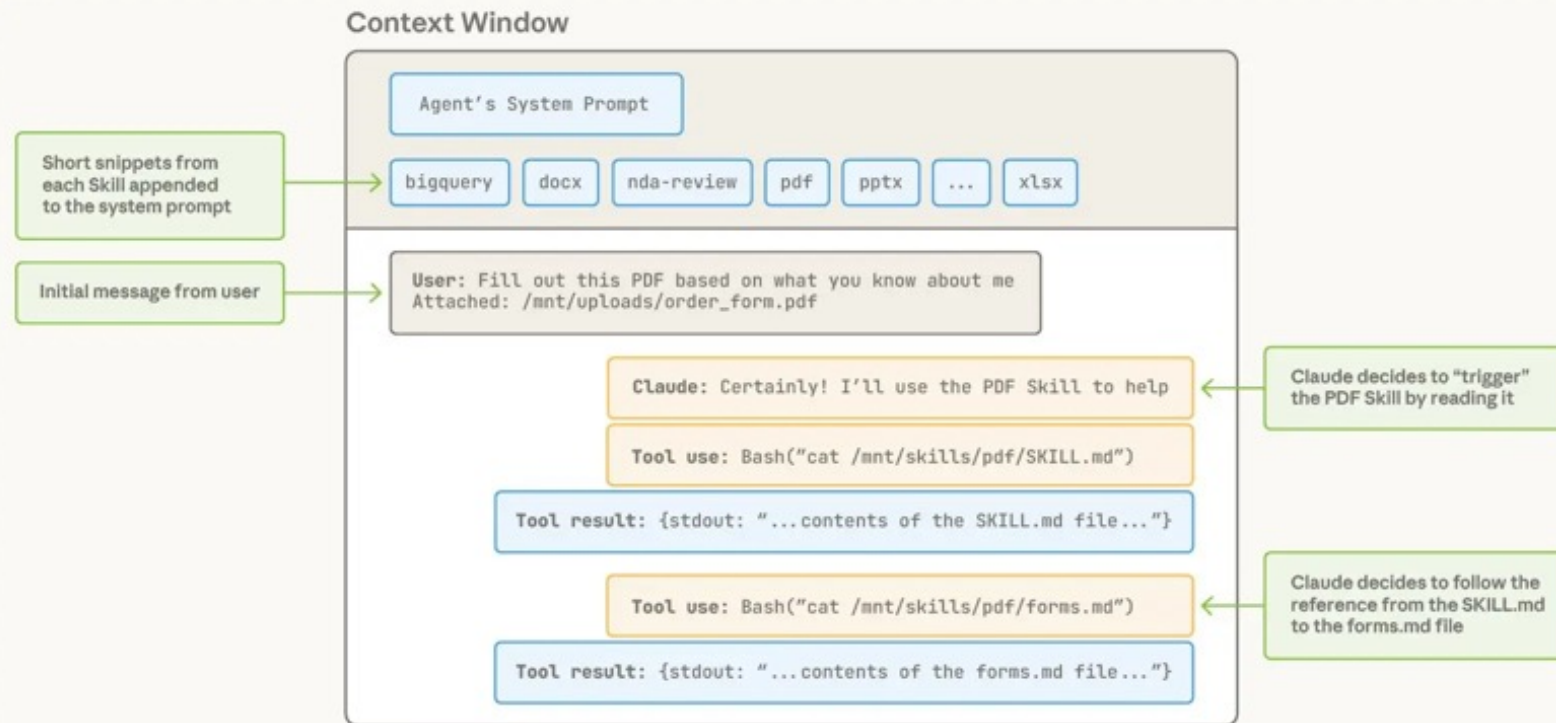
- Run this script from this file's directory:

```
`python scripts/extract_form_field_info.py <input.pdf>
<fields.json>`.
```

<https://github.com/anthropics/claude-cookbooks/tree/main/skills>

Claude Skills

Skills and the Context Window



Claude Skills

Bundling executable scripts

pdf/forms.md

If you need to fill out a PDF form, first check to see if the PDF has fillable form fields. Run this script from this file's directory: ``python scripts/check_fillable_fields <file.pdf>``, and depending on the result go to either the "Fillable fields" or "Non-fillable fields" and follow those instructions.

Fillable fields If the PDF has fillable form fields:

- Run this script from this file's directory:

``python ./extract_fields.py <input.pdf> <fields.json>``.

...

pdf/extract_fields.py

```
from pypdf import PdfReader

def write_field_info(pdf_path: str, output_path: str):
    """Extract form fields from PDF and store as JSON."""
    reader = PdfReader(pdf_path)
    fields = get_fields(reader)
    with open(output_path, "w") as f:
        json.dump(fields, f)

# ... omitted ...

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print(f"Usage: python {sys.argv[0]} <pdf_path> <output_json_path>")
        sys.exit(1)
    write_field_info(sys.argv[1], sys.argv[2])
```

Próximos pasos

- Realizar una propuesta/prueba de concepto
 - Identificar scope y requerimientos
 - Diseñar la solución (RAG, agentes, workflows, bases de datos, front?)
 - Tecnologías (LLMs, no necesariamente Python o Langchain)
 - Contar con datos (pueden ser semi-sintéticos) para probar
 - Implementar un prototipo que sea demostrable (opcional)



Gracias!



Andres Diaz Pace

andres.diazpace@isistan.unicen.edu.ar