BOOK-M@TE

Building an AI-assistant

About me

nupsea.github.io

Anup Sethuram

Senior Data-ML Engineer

WHY?

- Hard to remember
- Summarization & Insights
- Multi book Comparison
- Self - hosted

AGENDA

Chapter 1 – Foundations

- LLMs
- RAG & Search

Chapter 2 – AI Assistant

- Agents & MCP
- Evals & Monitoring

Chapter 3 – Book-mate App

- Architecture & Components
- Demo

Chapter 4 – Future

- Learnings, Next Steps
- Q n A

Chapter 1

FOUNDATIONS

LLMs

Strengths

- Natural language generation
- Reasoning and summarization
- Fast inference with modern models

Limitations

- Missing Context
- Outdated knowledge
- Hallucinations
- Token window limits

LLMs

Foundational Models (*Innovation*)

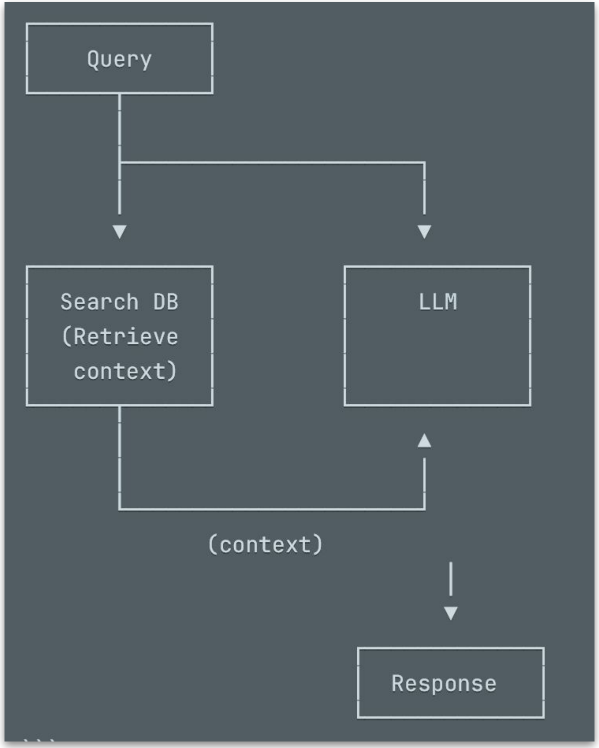
Pretraining

Adoption Approaches (*Engineering*)

- Fine-tuning: Adapt to your domain
- RAGs + Prompt Engineering + Agentic : Widely used

RAG

Retrieval
Augmented
Generation



Query

Search DB (Retrieve context)

LLM

(context)

Response

Advantages:

- Simple architecture and safe
- No retraining required
- Adds latest knowledge

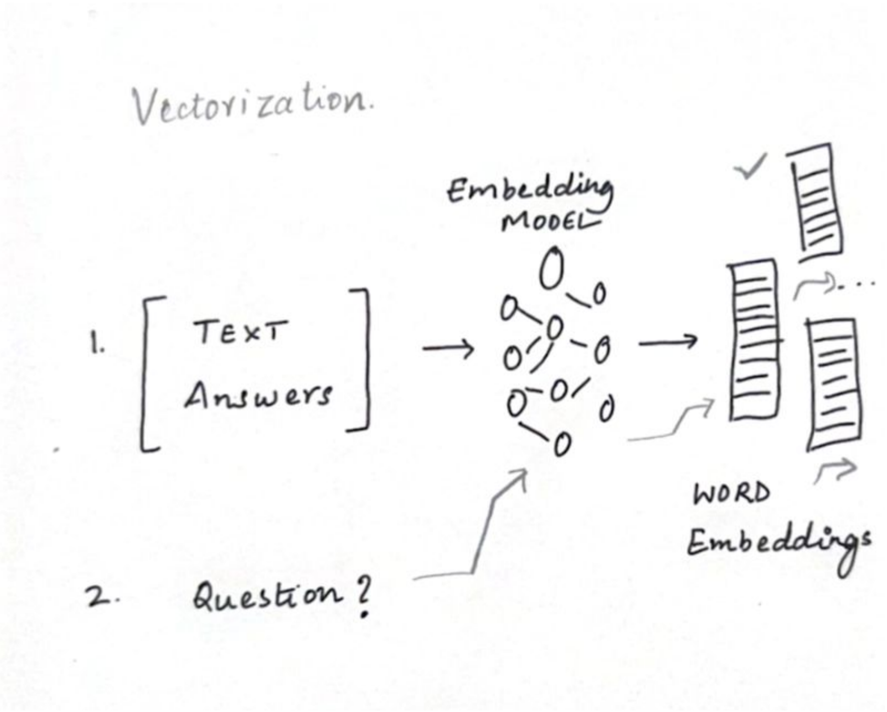
Challenges:

- Limited to the model’s skills.
- Heavy reliance on search.

Search

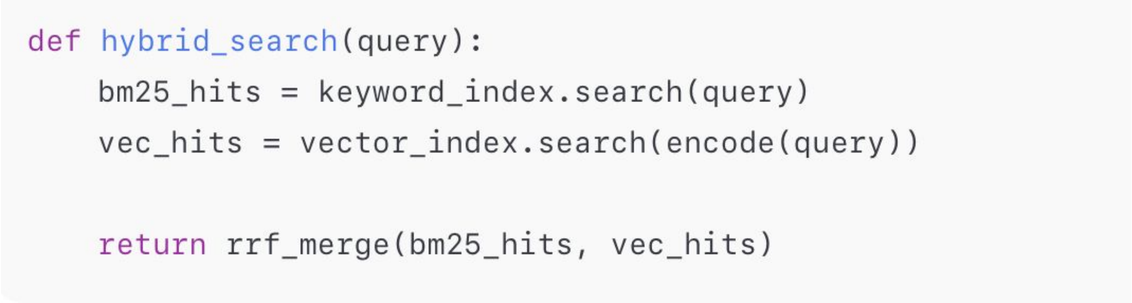
Query: "Who is Cheshire?"

- ❌ **Keyword search:** Fails if user says "grinning cat"
- ✅ **Vector search:** Understands semantic meaning



Search

- ✅✅ **Hybrid: Best of both**



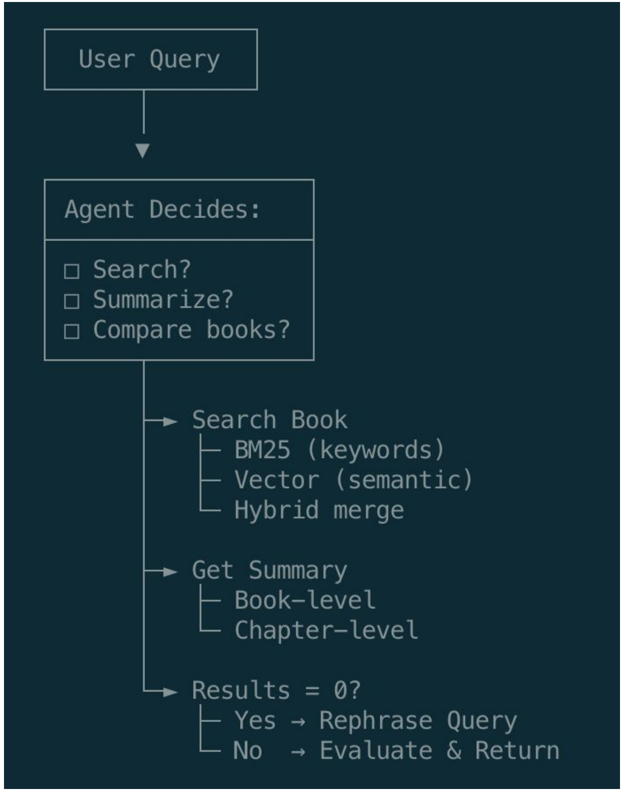
```python
def hybrid_search(query):
    bm25_hits = keyword_index.search(query)
    vec_hits = vector_index.search(encode(query))

    return rrf_merge(bm25_hits, vec_hits)
```

CHAPTER 2

ASSISTANT

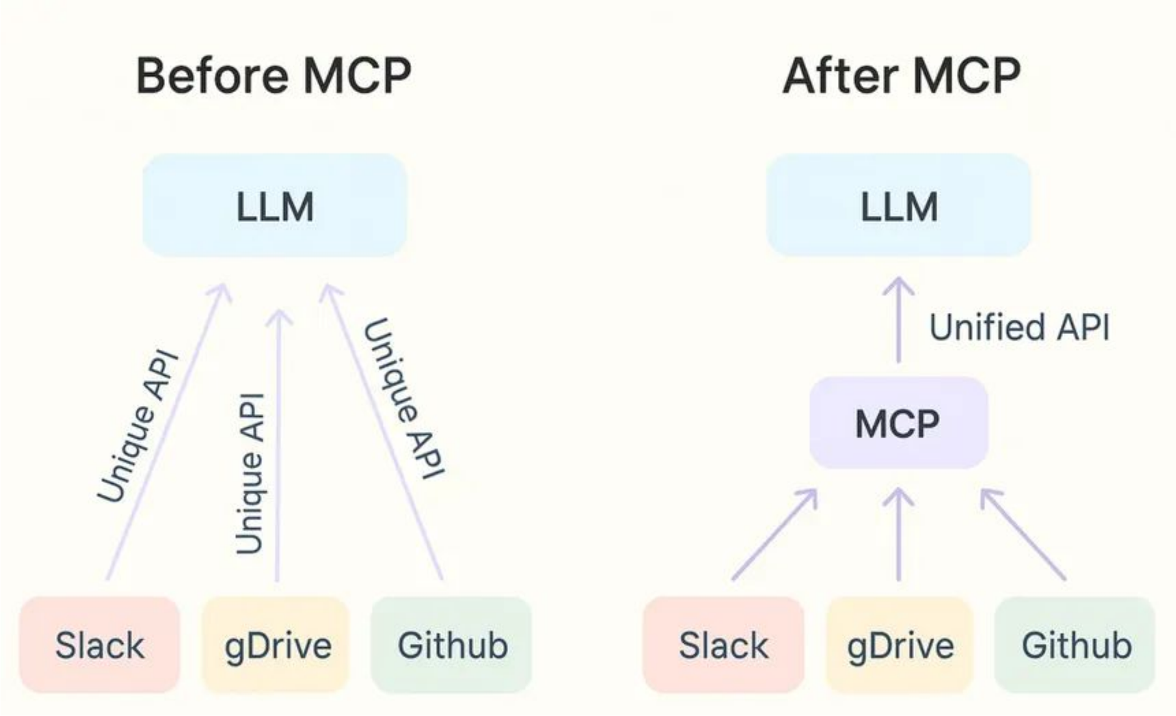
Agents

- ➢ Complex, multi steps
- ➢ External integration
- ➢ Dynamic Decision Making
- ➢ Accuracy & Reliability
- ➢ Stateful Operations
- ➢ Task Automation

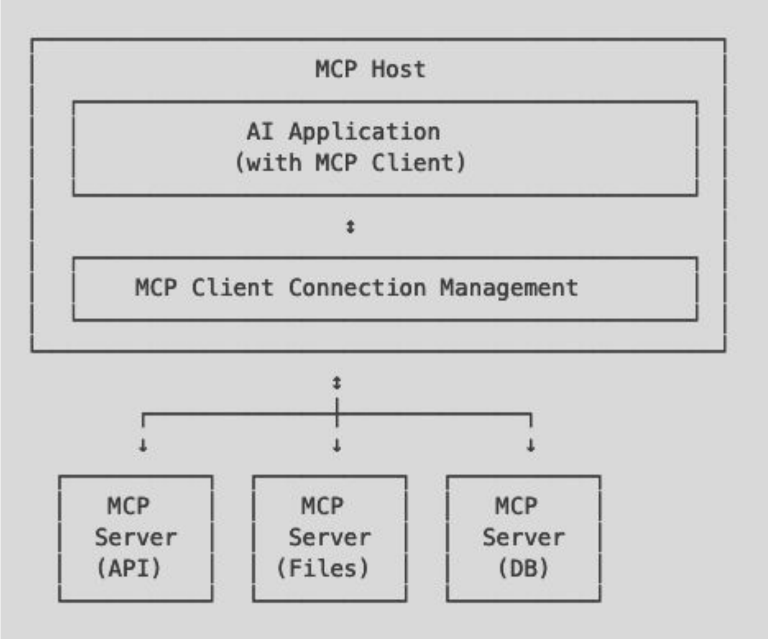


```
┌──────────────┐
│  User Query  │
└──────┬───────┘
       │
       ▼
┌─────────────────┐
│ Agent Decides:  │
├─────────────────┤
│ □ Search?       │
│ □ Summarize?    │
│ □ Compare books?│
└──────┬──────────┘
       │
       ├──► Search Book
       │    ├─ BM25 (keywords)
       │    ├─ Vector (semantic)
       │    └─ Hybrid merge
       │
       ├──► Get Summary
       │    ├─ Book-level
       │    └─ Chapter-level
       │
       └──► Results = 0?
            ├─ Yes → Rephrase Query
            └─ No  → Evaluate & Return
```

Model Context Protocol



MCP architecture



MCP Host

AI Application
(with MCP Client)

MCP Client Connection Management

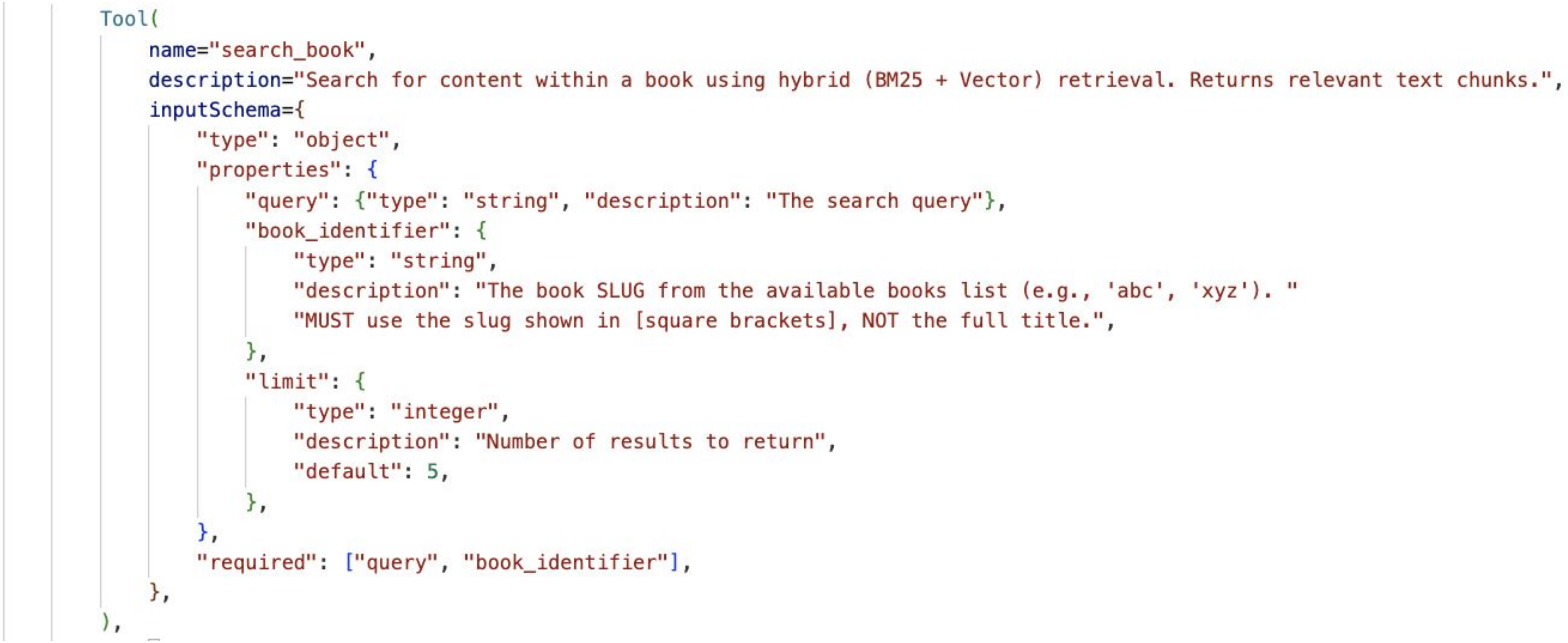
MCP
Server
(API)

MCP
Server
(Files)

MCP
Server
(DB)

- **MCP Server**: Exposes Tools [search_book, get_summary..]
  - Resources
  - Prompts
- **MCP Client**: Agent (makes decisions)
- **MCP Host**: Manages connections

MCP tools



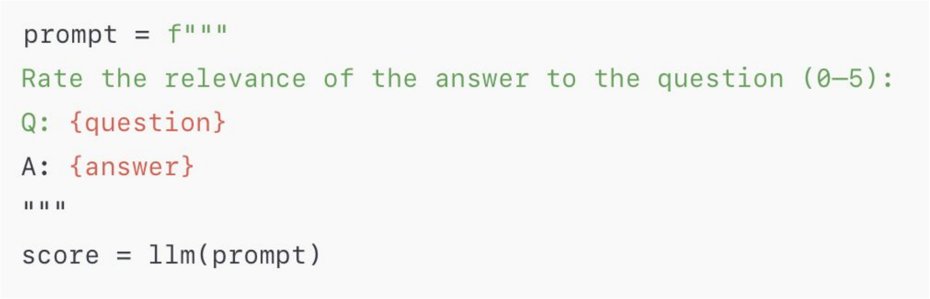
```
Tool(
    name="search_book",
    description="Search for content within a book using hybrid (BM25 + Vector) retrieval. Returns relevant text chunks.",
    inputSchema={
        "type": "object",
        "properties": {
            "query": {"type": "string", "description": "The search query"},
            "book_identifier": {
                "type": "string",
                "description": "The book SLUG from the available books list (e.g., 'abc', 'xyz'). "
                "MUST use the slug shown in [square brackets], NOT the full title.",
            },
            "limit": {
                "type": "integer",
                "description": "Number of results to return",
                "default": 5,
            },
        },
        "required": ["query", "book_identifier"],
    },
),
```

Eval & Monitoring

Search Eval: Golden data + HR & MRR

Book	Hit Rate@5	MRR@5
Hegel	70.3%	0.58
Marcus	68.1%	0.55

LLM Response Eval: LLM as a judge.



```
prompt = f"""
Rate the relevance of the answer to the question (0–5):
Q: {question}
A: {answer}
"""
score = llm(prompt)
```

Observability:

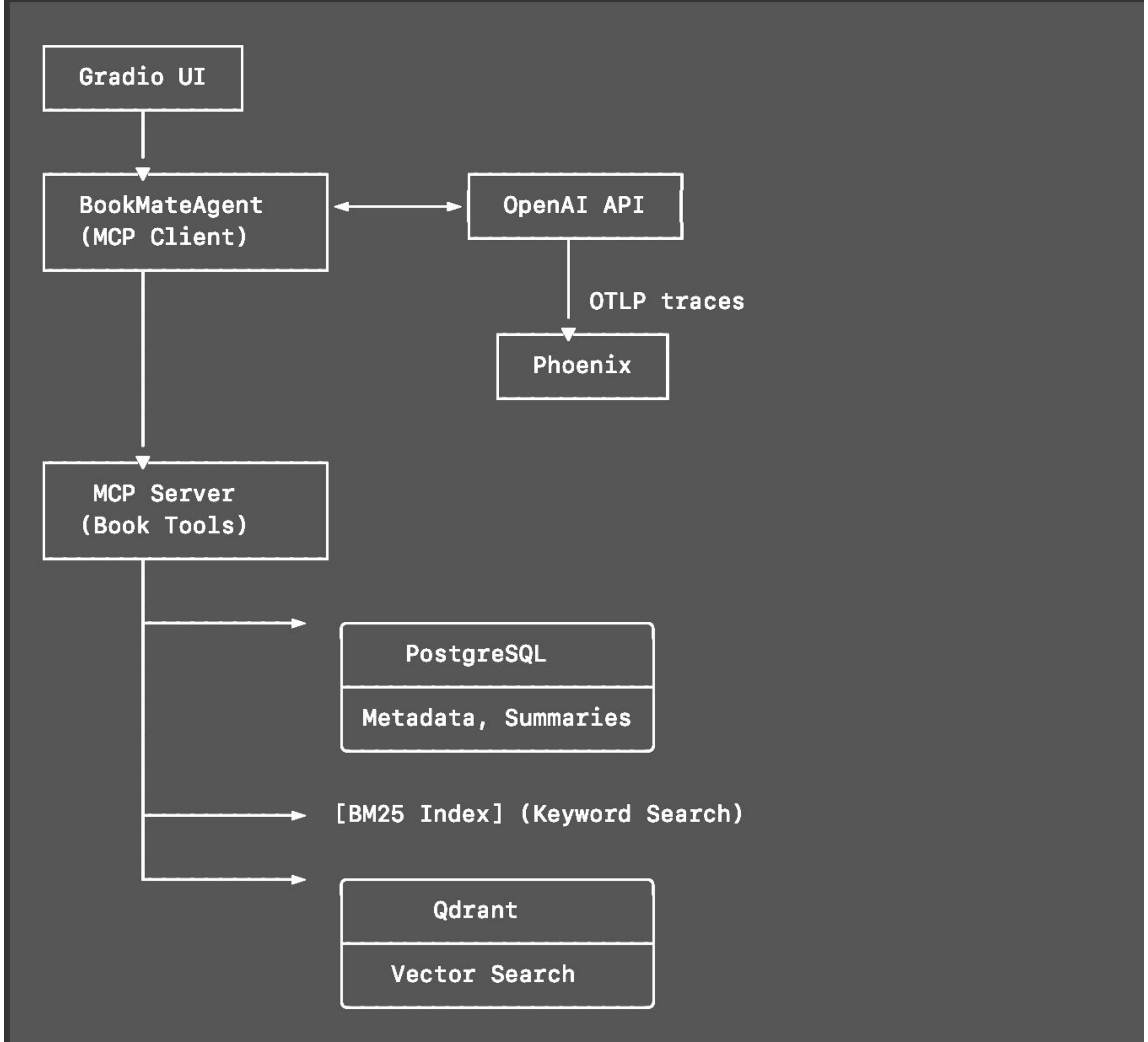
- ***Metrics*** with User Ratings
- Distributed ***Tracing***
- Automated Tests

Chapter 3

Book-mate app

Architecture

book -mate



Gradio UI

BookMateAgent
(MCP Client)

OpenAI API

OTLP traces

Phoenix

MCP Server
(Book Tools)

PostgreSQL

Metadata, Summaries

[BM25 Index] (Keyword Search)

Qdrant

Vector Search

DEMO

Chapter 4

Future

Learnings & Challenges

- Search optimizations to improve eval metrics
- Non deterministic workflows
- Real world: different book sources.

Next Steps:

➢ Supporting more formats and expanding ecosystem.
➢ Security and Essential Guard Rails with Alerting
➢ Comprehend images and extend to technical-docs
➢ Data collection and analytics.

Q&A

Suggestions:

- ➢ Try it yourself. github.com/nupsea/book-mate
- ➢ ⭐ the repo if you found it useful
- ➢ Highly appreciate contributions

References:

- ➢ LLM Zoomcamp - Alexey

  datatalks.club/courses/llm-zoomcamp
- ➢ Books - Gutenberg Project

  https://www.gutenberg.org/

Thank you!

Anup Sethuram

Senior Data-ML Engineer