



Orchestrating AI Knowledge Bases with Apache Airflow

3.0

Theo Lebrun, Senior Data Engineer, Ippon
Technologies

Agenda



PART 01

Intro

What is Generative AI?

PART 02

Knowledge Bases?

Orchestration with Airflow

PART 03

Best Practices & Tips

Conclusion and Q&A

Theo Lebrun



lebruntheo



Falydoor



@Falydoor

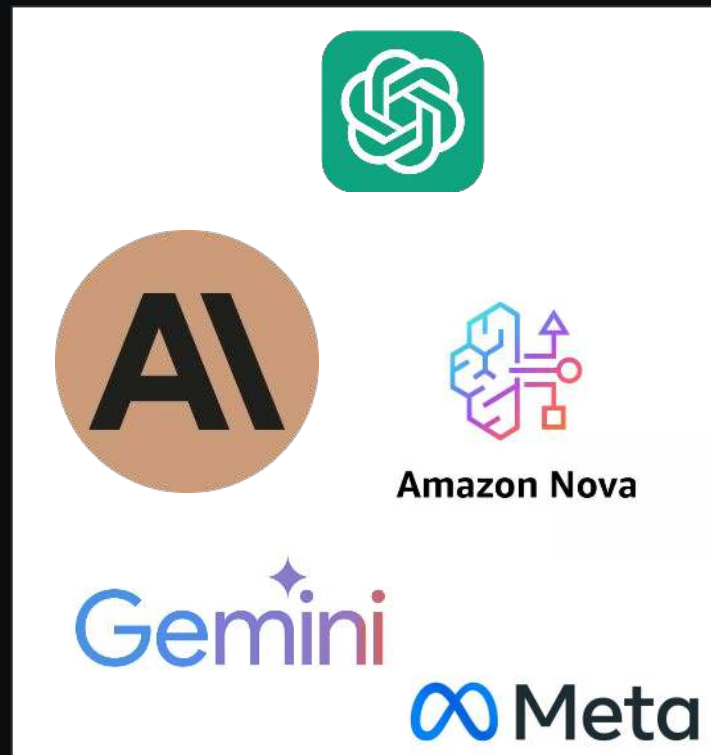
<https://theolebrun.com/>

Senior Data Engineer and
Technical Manager at [Ippon](#)
[Technologies](#)

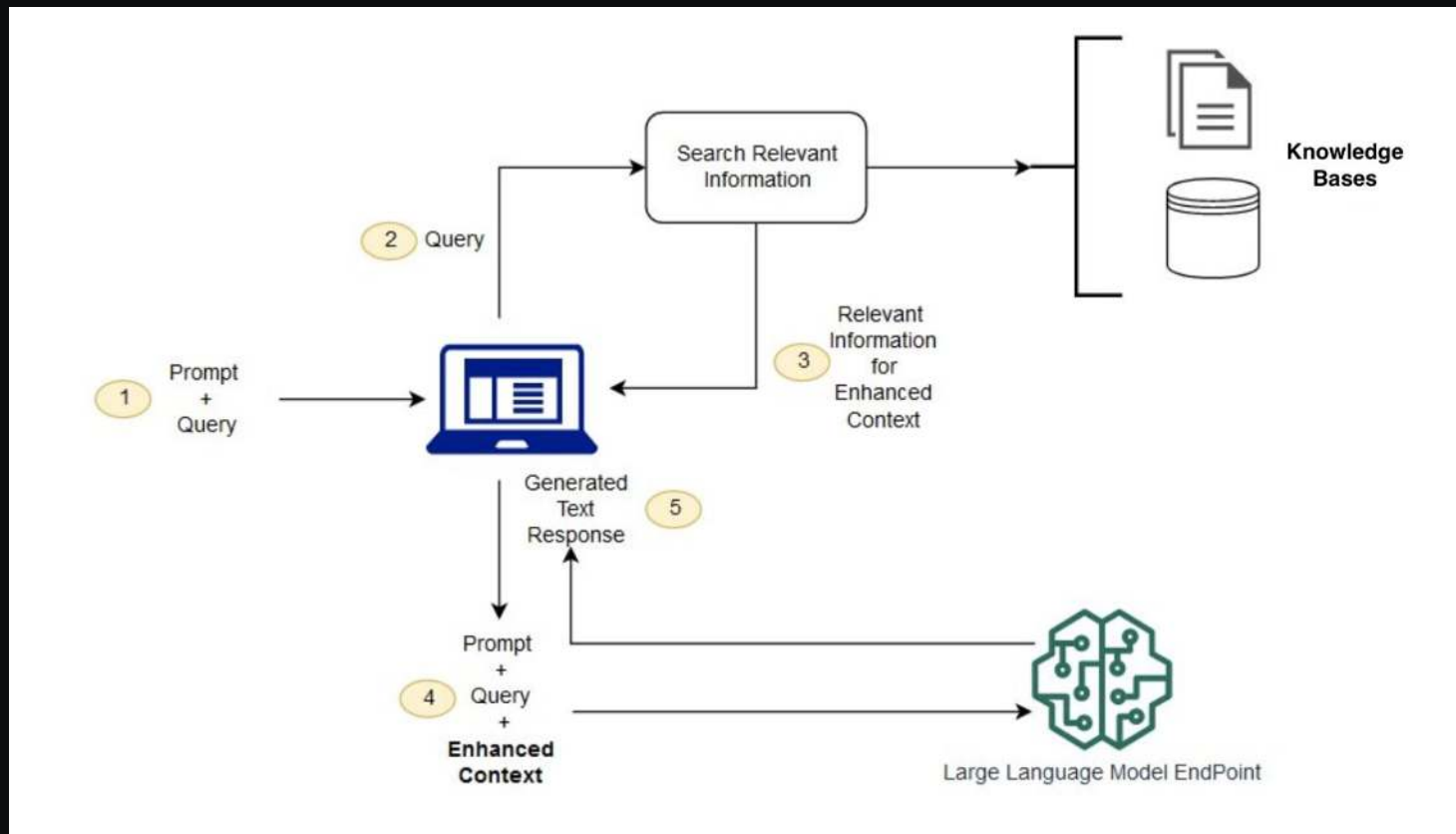


What is Generative AI?

- Generate Human-like Text
 - Based on a query (or prompt)
 - Can be enhanced with a context (RAG)
- Tailored
 - Model provider
 - Model type
 - Temperature
- Use Cases
 - Customer support
 - Content creation
 - Virtual assistants
 - And more...



Retrieval-Augmented Generation (RAG)



IpponAssistant



- GenAI Slack Chatbot
- Knowledge Bases
 - Resumes
 - Case studies
 - Internal docs (handbook, healthcare benefits, etc)



Théo LEBRUN 18:45

Who has Snowflake experience?



IpponAssistant APP 18:45

Based on the context documents provided, the following individuals at Ippon have experience with Snowflake:

- Stephen [REDACTED] - Has SnowPro Core Certification and experience architecting and implementing Snowflake data warehouses for clients.
- Christopher [REDACTED] - Has Snowflake Professional Architect certification and experience leading Snowflake implementations for clients.
- Pooja [REDACTED] - Developed Python jobs to replicate data from various sources into Snowflake using Airflow and has experience maintaining and contributing to data architecture in Snowflake.
- Nicolas [REDACTED] - Has experience managing Snowflake implementations and migrations as part of his previous role as Business Manager at Ippon.

I did not see any other mentions of individuals having specific Snowflake experience in the context documents provided. Let me know if you need any additional details about the Snowflake experience of these individuals. (edited)

Knowledge Bases

- Sources

- File based
- Database
- Third party



- Data Requirements

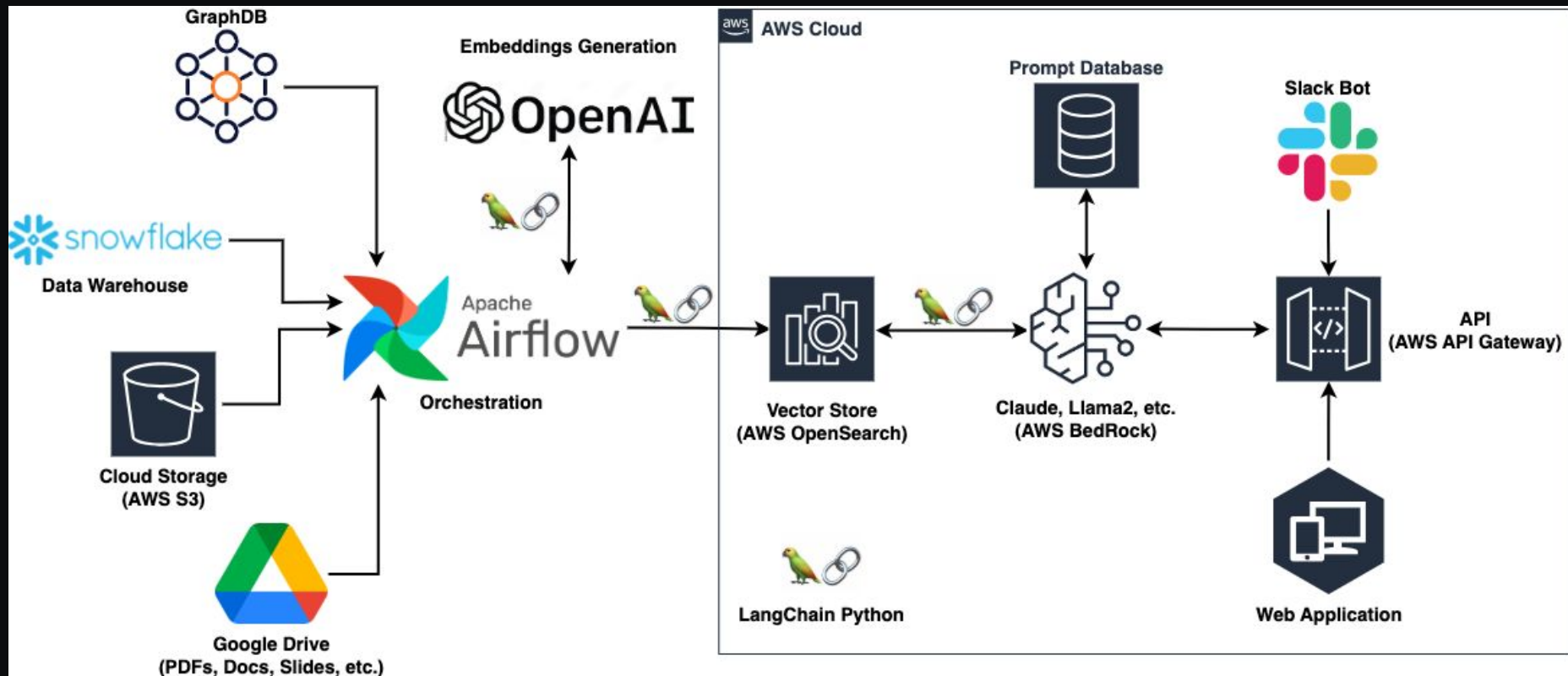
- Collect, clean and standardize
- Update frequency
- Metadata and tagging



- Document Integration

- Type of documents (doc, pdf, txt, etc.)
- Chunking strategies

Architecture

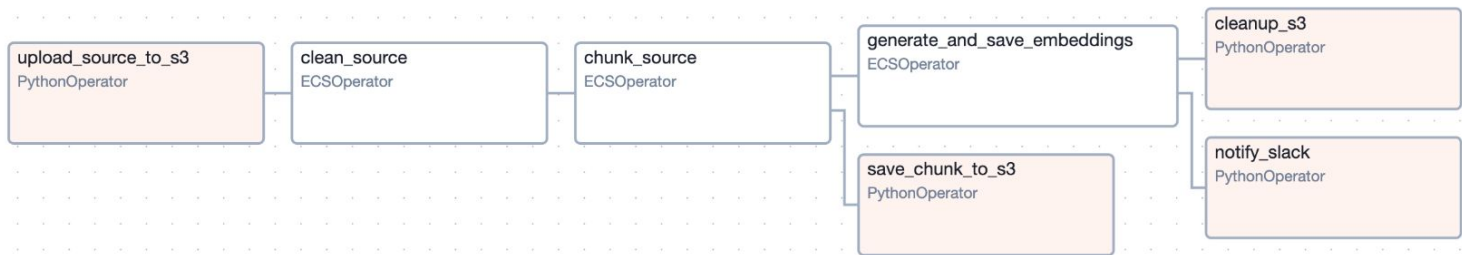


DAG Example



△ Details **Graph** 📅 Gantt <> Code 📅 Event Log ⌚ Run Duration ⌚ Task Duration 📅 Calendar

Layout:
Left -> Right ▼



React Flow

Tips and Tricks

- Dag Processing
 - Delegate data processing (ECS, k8s pod operator, etc.)
 - Dags should be idempotent
 - Atomize your tasks
- Knowledge Base
 - Multi index and prompt (reduce LLM hallucination)
 - Prepare, clean and standardize your data
 - Add metadata to your documents

Thank You for Your Time!

I would love to chat more with you about using Airflow to build GenAI applications.

You can email me at tlebrun@ipponusa.com or message me on



Questions?