# GENERATIVE AI'S POTENTIAL FOR KNOWLEDGE MANAGEMENT

**In the first part of our generative AI's series, we explored how GenAI can industrialize the software delivery process. During our experiments, it became clear that the quality and detail of the input queries – the natural language requests associated with writing code, creating user requirements, testing and so on that we presented to the AI models – strongly affect the quality of the output generated. In this second article, we focus on the AI's potential to aggregate knowledge and generate reliable and detailed responses to natural language queries.**

## EXPERIMENT OVERVIEW

To allow us to test these concepts, we chose to use a knowledge management system that captures information around the technology stack supporting a Tier 1 bank's instant payments solution.

More specifically, the information in this knowledge management system is stored as a knowledge graph, a type of database which allows the source information to be captured in a highly structured manner. The more structured the source information, the more accurate and contextually correct the AI's responses will be.

Once the AI has been provided with contextual information around the architecture and technology stack of our instant payments solution, we presented it with complex questions on various topics (see examples in the table below).

| TOPIC | QUESTION |
|---|---|
| **Knowledge transfer** | Which microservices and API endpoints form part of the customer onboarding journey, and how do they interact as part of the onboarding workflow? |
| **Impact analysis** | A data element commonly used across services will change from numeric to alphanumeric. Which API endpoints across the entire services stack are impacted? |

| TOPIC | QUESTION |
|---|---|
| **Consolidation** | I am working on a strategy to deduplicate technology solutions in the bank. Show me how the technical solution for obtaining an account statement varies between retail and SME customer segments? |
| **Design** | Based on the current architecture that supports the instant payments business capability, where will I have to modify my existing technology stack to future-date the instant payment? Specify which data elements I would need to include on new or modified API endpoints. |

# MAGNITUDE OF THE TASK

For a complex banking application, such questions are not easy to answer, as the information would need to be pulled from many different source documents, and aggregated and analyzed to derive the correct response.

A typical banking app comprises hundreds of interconnected components. Answering our questions requires access to the technical specs for each of these components, ensuring that each of these specs is up-to-date, and then processing this vast amount of information to formulate a response. This is an immense task that is as good as impossible for a human to complete.

AI can remove the burden of knowledge aggregation and dissemination, taking on the task of updating information in a knowledge system, and generating answers in response to natural language questions and requests.

# HOW AI PERFORMED

As part of our experiment, we created a Python script to convert the entire instant payments app technical knowledge database from endless pages of raw code – which would be inefficient and expensive for the AI to work with – into a 'narrative', a set of natural language statements which provide semantic context for the AI.

We then tested the AI's ability to respond to our natural language questions and requests, using the narrative version of the database as its source material.

On the scale of one to five we adopted in the first part of this series, where 'five' represents AI outputs that can be fully trusted and used without any human oversight or review, we saw a fundamental improvement – from 3 to 4.5 – in the quality of the AI's responses compared to the results we described in that previous article.

The results returned by GenAI in response to straight-forward questions or requests – for example, **'Which data element is most frequently used across all API endpoints, and what is its purpose?'** or **'Can you create a component diagram of the solution that describes the input I provided?'** – were 100 percent accurate. This is not surprising considering that data extraction and summarization are GenAI's key strengths.

Moreover, the AI was able to provide highly detailed answers to questions that have a greater degree of abstraction – for example, **'Do you think the microservices that were mapped are well isolated, and why?'**.

We were further impressed by the level of accuracy and detail in the AI's responses when asked to create the app design assets based on new requirements – for example, **'Can I reinstate a cancelled future-dated payment? If not, how would I modify existing endpoints or create a new endpoint to achieve this? Please provide me with detailed API specifications.'**

Finally, the AI provided valid responses to queries about the app architecture convergence, based on its assessment of similarities between different parts of the app design – for example, **'Which two API endpoints are most alike in their response definitions, and why? Can they be converged to a single endpoint?'** AI models identified code duplication, system inefficiencies and opportunities for functionality consolidation.

# CONCLUSION

The AI's responses based on structured and comprehensive contextual information were accurate, detailed and complete. We would feel confident in using GenAI for knowledge transfer, impact analysis, architecture consolidation and solution design related activities – the tasks that banking IT specialists wrestle with as part of day-to-day technology support.

Armed with quality source data which is highly structured (as in our example of using a knowledge graph), GenAI acts as a dynamic knowledge engine that can provide reliable answers to complex technology related questions, in seconds.

The use of GenAI to update, manipulate and disseminate information in a knowledge management system is not limited to our instant payments app use case. Any knowledge system where complex information can be effectively modelled will benefit from this approach. Examples include:

- Supporting fraud detection, risk assessment, and compliance by analyzing relationships between entities such as customers, transactions, accounts, and external data sources (e.g. market data, credit reference agencies, sanctions lists).

- Powering recommendation systems and suggesting products and services by analyzing the relationships between user needs and preferences and transaction data.


## AUTHORS

**Gerhardt Scriven,** Managing Principal

**Rodrigo Sarai,** Principal Consultant

**Marcel Braga,** Principal Consultant

**Jamesson Jr,** Senior Consultant

## CONTACTS

**Alessandro Corsi,** Partner, alessandro.corsi@capco.com

**Luciano Sobral,** Partner, luciano.sobral@capco.com


WWW.CAPCO.COM

CAPCO
a wipro company