ChatGPT for Microservice Development: How far can we go?

Lauren Adams[1] · Francis Boyle[1] · Patrick Boyle[1] ·
Dario Amoroso d'Aragona [2] · Tomas Cerny[1] · Davide Taibi[2,3]

[1] Baylor University, Waco (United States)
[2] Tampere University, Tampere (Finland)
[3] University of Oulu, Oulu (Finland)

✉ lauren_adams3@baylor.edu · francis_boyle1@baylor.edu · patrick_boyle1@baylor.edu ·
dario.amorosodaragona@tuni.fi · tomas_cerny@baylor.edu · davide.taibi@oulu.fi

1 Introduction

Both microservices and generative AI play significant roles in code development, prompting the question of how generative AI can aid in building a microservice-based system. Although *ChatGPT-4* has been evaluated for structuring microservices and generating code snippets, limited research exists on its ability to fully develop a microservice-based system. This paper aims to evaluate whether *ChatGPT-4* can construct a complete system with minimal human oversight, using a known development example. To address this objective, we formulated the following research questions (**RQ**s):

RQ$_1$ *Is it possible to fully build a microservices system only using ChatGPT?*

RQ$_2$ *What are the limitations of ChatGPT for building microservice architecture?*

RQ$_3$ *What manual/human interventions are needed?*

To achieve this, we conducted a preliminary investigation by attempting to replicate SockShop[1], a demonstration example of microservices systems, through conversation with *ChatGPT-4*. The result of this paper will enable a new line of research into the automatic generation of new services or automated software maintenance. The remainder of this paper presents related works, the approach adopted for building the system, a research roadmap towards automated generative microservice-based development.

2 Background and Related Works

2.1 Generative AI Tools

Generative AI is the focus of creating new and unique content with trained data [2]. This content can be very versatile, stretching from literature, images, music, academic papers, or even films. Generative AI utilizes Natural Language Processing (NLP), which has the capabilities of translating speech

[1] https://microservices-demo.github.io/

to text, tagging parts of speech within the text, recognizing named entities, extracting sentiment or emotions from text, and generating natural language [2]. Various generative AI programs have been released on the market, the most well-known of these being ChatGPT. We will move on to discuss the capabilities of *ChatGPT-4* and its various competitors.

Companies leading in the field of generative AI and actively developing chatbots include Google, Meta, DeepMind, and OpenAI [2]. The successful release of ChatGPT resulted in a rush of AI development and releases and a race in AI development across the globe. After the release of major LLM Competitors soon followed [7] including:

- ChatGPT-3.5 - OpenAI released on November 22nd, 2022 as an advanced natural language processing model that has been pre-trained on a large dataset so that it can generate content that is contextually appropriate and at times indistinguishable from human-written content [2]. It became a cultural sensation with a record-breaking estimated 123 million monthly users after only 3 months of release, making it the fastest-growing application of all time [7]. Specifically, it has been trained through Reinforcement Learning from Human Feedback (RLHF) to follow instruction in a prompt and provide a detailed response [5]. Some limitations include occasionally writing nonsensical or implausible answers, inability to ask clarifying questions and only guessing intent, failure to successfully detect unsafe or harmful instructions, excessive verbosity, and sensitivity to tweaks in input phrasing [5].

- Bard - Alphabet (Google Parent company) released on February 6th, 2023 as a companion to the search engine and as an experimental demo. Notably, the ability to generate computer code has been turned off.

- Bing Chat - Microsoft released on February 7th, 2023. An integration of ChatGPT into the Bing Search engine. Unlike ChatGPT, it has access to the internet and can give proper citations for the sources it uses, but currently has conversation limits.

- GPT-4 - OpenAI released on March 14th, 2023 with more advanced capabilities than GPT-3.5. Both GPT-4 and GPT-3.5 were evaluated on their performance in a set of 34 technical evaluations originally intended for humans, including the LSAT, GRE, various AP exams, and Leetcode evaluations. In these evaluations, GPT-4 outperformed GPT-3.5 on the majority of the given evaluations. Major advancements of GPT-4 include capabilities to receive visual and textual inputs, and it followed a predictable scaling model, giving its creators much greater capacity to predict how it will develop and respond [5].

- Ernie - released by Baidu on March 16th, 2023. Trained on a greater dataset than GPT-3 with 260 billion parameters to GPT-3's 175 billion. Developed with a focus on enterprise abilities rather than engaging the general public and is trained to have a much greater understanding of Chinese culture and the ability to comprehend various Chinese languages.

2.2 Related Work

In this section, we present an overview of existing studies focusing on the use of AI in software architecture and functional systems. Stojanovic et al. [8] explore ChatGPT's ability to analyze software requirements and identify microservices in different systems. Detailed and accurate information is found to be crucial for obtaining high-quality solutions, highlighting ChatGPT as a helpful tool rather than a replacement for software architects.

Cámara et al. [3] investigate ChatGPT's performance in generating syntactically and semantically correct UML models. While ChatGPT can produce generally correct models, it exhibits syntactic deficiencies, lack of consistency, and scalability issues.

Ozkaya [6] examines the implications of large language models (LLMs) in software engineering. LLMs can assist in tasks like specification generation, developer feedback, testing, documentation, and translation. However, concerns regarding data quality, bias, privacy, environmental impact, explainability, and unintended consequences need to be addressed.

Ahmad et al. [1] explore ChatGPT's role as a software development bot, collaborating with human architects to automate the process of architecting software systems. While ChatGPT can generate architectural artifacts and provide decision support, it requires human oversight.

Dae-Kyoo [4] highlights ChatGPT's capabilities in various software development tasks. However, limitations related to traceability and artifact consistency necessitate human intervention. Ongoing training may mitigate these limitations, but human oversight remains crucial for ensuring high-quality outputs.

Combining the expertise of human developers with ChatGPT's abilities can make it a valuable assistant in software development.

3 Building the E-commerce App

In this Section, we report our approach for ChatGPT-led development Section 3.1 and the application of the approach to the specific case study Section 3.2 and the specific features and system structure implemented Section 3.1

3.1 Method

The study was performed in five steps:

Step 1: Project Selection: Define the criteria to select a project to replicate. In particular, we established the following criteria: complexity, feasibility, sound construction, and level of documentation, in order to ensure a properly constructed analysis.

Step 2: Requirement Elicitation: After the project selection, we studied the system and extracted the requirements of the system.

Step 3: Develop Prompts: The requirements were converted into specific prompts for *ChatGPT-4*.

Step 4: Generate Code: Starting from the requirements we started to chat with *ChatGPT-4* asking it to develop the code.

Step 5: Review: Validate the code, paste it into the appropriate locations within systems as specified by the *ChatGPT-4* instructions, and prompt the next requirement or improvement.

Note that this is an iterative process because *ChatGPT-4* cannot provide all the code and often *ChatGPT-4* is needed to fix errors in the generated code. This process is described in Figure 1. Once the systems were complete, we evaluated the systems to categorize what types of errors were present in the code, what issues occurred in the generation process, and what skills are needed to develop these types of systems into functional microservice systems.

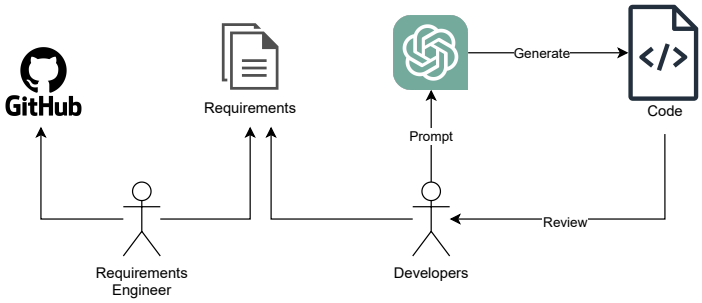


Figure 1: This diagram demonstrates the process used for this experiment.

3.2 Conducting the Study

As for project selection, we evaluated a microservice demo application called SockShop and selected it due to its comprehensive documentation, compact size, and ability to represent the complexities of a microservice system. Moreover, it was developed as a demonstration of microservice architecture, making it an appropriate model for our analysis. SockShop allows users to shop for and buy socks and is implemented with three key microservices: User, Catalog, and Cart. We analyzed the functionality of the components, translated them into prompts, and ran them through *ChatGPT-4*.

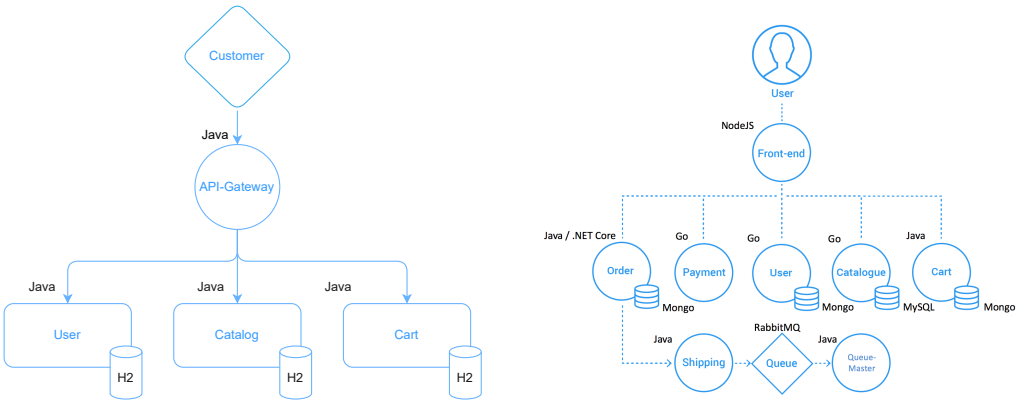
We opted to simplify the structure by focusing on the core services that showcased the interrelation between microservices and encompassed most of the functionality of SockShop. We implemented an API-Gateway to access the services. For simplicity reasons, we did not implement the frontend. Additionally, for consistency throughout the experiment, we implemented all microservices using Java Spring Boot and H2 databases, which could be easily configured by *ChatGPT-4*. The package structure generated by *ChatGPT-4* for all services followed the pattern: model, repository, controller, service. Within those classes, the code generated followed the typical SpringBoot coding conventions and annotations. The complete list of requirements, prompts, and the generated code can be viewed in more detail at the Online Appendix [2]

4 Results and Discussion

Our findings (**RQ1**) indicate that relying solely on *ChatGPT-4* for constructing a microservice application poses significant challenges. While *ChatGPT-4* can generate code for different components, human intervention is essential for ensuring successful application execution. However, building a small application with *ChatGPT-4* assuming the majority of code generation responsibility is feasible, as demonstrated in the case study.

The limitations of using *ChatGPT-4* for microservice architecture development (**RQ2**) include limited conversational memory, outdated information, and its chatbot nature. *ChatGPT-4* is effective in addressing requirements with a well-defined system structure and detailed specifications. However, it struggles with accommodating changes or handling debugging scenarios, occasionally suggesting solutions that deviate from the established architecture. Its scalability is limited to specific chats, requiring constant context reminders for accurate code generation. Furthermore, *ChatGPT-4* may suggest outdated packages or dependencies due to the dataset cutoff point. It also leaves gaps for developers to implement logic and security, as it was not designed as a developer tool. These limitations

[2]https://github.com/M3S-Cloud/ChatGPT4MS



(a) The implemented Architecture

(b) SockShop Architecture

Figure 2: Architecture of the systems

hinder the effectiveness of *ChatGPT-4* in constructing a microservice architecture.

Human intervention occurred in two major stages during the experiment (**RQ3**). The initial phase involved breaking down the system into prompts for *ChatGPT-4*, requiring individuals with a comprehensive understanding of the target system and software development principles. The success of this stage greatly impacts the amount of corrections needed before deployment. The subsequent phase involved debugging and deploying the application, where *ChatGPT-4* provides assistance but human intervention remains indispensable. The complexity of the system determines the required expertise, and in some cases, human developers prove more efficient in resolving issues compared to relying solely on *ChatGPT-4*.

5 Research Roadmap

As research progresses, we aim to test the limits of developing with minimal human intervention. An example of this is the creation of a deployment pipeline using *ChatGPT-4* to automate the deployment process. We also intend to explore the extension and integration of generative AI into an Integrated Development Environment (IDE), similar to GitHub's Copilot. Another significant step is the validation of systems constructed by *ChatGPT-4*. This includes testing functional and non-functional requirements, implementing automated testing systems, and conducting a qualitative assessment of the architecture by a professional system architect. Furthermore, we plan to expand this study to encompass other generative AI technologies such as Ernie, Bing Chat, or Bard.

6 Conclusion

In this work, we conducted a preliminary investigation to assess the capabilities of ChatGPT in generating a microservice-based system with minimal intervention.

The results demonstrate that ChatGPT can be a valuable support tool for microservice development. However, it requires intervention to ensure the accuracy and completeness of the generated code. Notably, ChatGPT occasionally forgot the code of previously developed services, necessitating careful verification by developers. Additionally, in some cases, *ChatGPT-4* recommended code

changes without directly applying them to the previously generated code.

It is important to acknowledge that *ChatGPT-4* is not primarily designed for application development, and using it for such purposes involves certain challenges. Due to its chatbot nature, the responses generated by *ChatGPT-4* can vary significantly, introducing an element of unpredictability when generating code. While *ChatGPT-4* can expedite the development process and perform a substantial amount of code generation when provided with appropriate prompts and monitoring, it currently cannot autonomously develop applications without human supervision.

Furthermore, *ChatGPT-4* struggles with complex or specific issues, making it more suitable for simpler tasks. It is worth mentioning that the automation of software maintenance could be the next area for generative AI tools to explore. Existing tools like Copilot have already made strides in this direction. However, significant research efforts are still required before fully automated development, without human intervention, can be considered a viable option.

This study represents a step forward in automated software development, and it opens up possibilities for future research in automated code generation and code maintenance tools. While the potential for such tools exists, the need for human expertise and oversight remains crucial in ensuring the quality and reliability of software systems.

References

[1] Aakash Ahmad, Muhammad Waseem, Peng Liang, Mahdi Fahmideh, Mst Shamima Aktar, and Tommi Mikkonen. Towards human-bot collaborative software architecting with chatgpt. In *International Conference on Evaluation and Assessment in Software Engineering*, EASE '23, 2023.

[2] Ömer Aydın and Enis Karaarslan. Is chatgpt leading generative ai? what is beyond expectations?, January 2023. `https://openai.com/blog/chatgpt`.

[3] Javier Cámara, Javier Troya, Lola Burgueño, and Antonio Vallecillo. On the assessment of generative ai in modeling tasks: an experience report with chatgpt and uml. *Software and Systems Modeling*, 22(3):781–793, 2023.

[4] Dae-Kyoo Kim. Using chatgpt to develop software systems: Alert to software engineers?, 04 2023. `https://www.researchgate.net/publication/369692992_Using_ChatGPT_to_Develop_Software_Systems_Alert_to_Software_Engineers`.

[5] OpenAI. Website, November 2022. `https://openai.com/blog/chatgpt`.

[6] Ipek Ozkaya. Application of large language models to software engineering tasks: Opportunities, risks, and implications. *IEEE Software*, 40(3):4–8, 2023.

[7] Jürgen Rudolph, Shannon Tan, and Samson Tan. War of the chatbots: Bard, bing chat, chatgpt, ernie and beyond. the new ai gold rush and its impact on higher education. *Journal of Applied Learning and Teaching*, 6(1), 2023.

[8] Tatjana Stojanovic and Saša D. Lazarević. The application of chatgpt for identification of microservices. *E-business technologies conference proceedings*, 3(1):99–105, Jun. 2023.