Project plan: AI project 9/10/2023

Description of the project?

The project is a study on the use of (generative) AI to improve the services of SSC ICT by supporting employees and optimizing internal processes. In particular, the focus is on generative large language models (LLM) because they can have the most significant impact on the daily work of SSC-ICT employees. The assignment (Speech Recognition & AI) has been approved by the SpB and has been running since the beginning of 2023 (internal hour code **201202044.002**).

Why now?

Generative AI is already being used by more than a quarter of the employees of the Dutch government. It has the potential to simplify parts of very diverse work and can save a lot of time. The flip side is that services are currently being used (ChatGPT, BingChat, Google) that save the data they are fed with, for further use. By providing paid AI services or developing services themselves, data streams can be controlled, and the use of AI remains safer. Furthermore, generative AI can be tailored to domain-specific tasks when set up by SSC-ICT itself.

Who will use it?

The intended users of the AI applications are all employees of SSC-ICT. For each sub-study, a specific use case will be developed for the SSC-ICT service desk.

How?

Several possible applications for AI are being explored. In this exploration, we will pay attention to alignment with existing projects, investigate existing AI tools, and research the possibility of developing AI services internally.

It is important to use AI responsibly. Therefore, all applications will be assessed for ethical responsibility, privacy, and compliance with AI-related laws and regulations.

Background Information

The innovation team conducted extensive research on machine learning and AI between 2016 and 2022. The conclusion was drawn that having its own AI implementation is technically feasible and highly desirable. Furthermore, multiple use cases for this technology were explored and identified as suitable candidates, including chatbots, documentation, and internal training material generators. In 2023, the service portfolio board asked the innovation team to conduct an AI quick scan and impact analysis, including the deployment of generative AI. This project builds upon the knowledge gained previously.

The Dutch government's attitude towards AI applications is somewhat ambivalent. On one hand, there is a demand, as many employees already use various (generative) AI services for their work. However, an approved offering is still scarce. This is partly due to the fact that there is division within the Dutch government on how to approach AI. Some see obstacles and would prefer to keep AI completely out of the picture. However, AI applications continue to grow, and it will be impossible to avoid participating in them in the future. It is important that the current project also addresses the impact of AI and the ethical issues it raises.

Sub-studies & Objectives

1. Development of an open-source chatbot
   1. Open-source Large Language Model (LLM) research

There are many open-source large language base models available. These models are often freely downloadable in various versions and sizes. This allows for running these models locally and experimenting with them. In the initial phase of this project, research was conducted on open-source LLMs to determine the different models available and which ones are most suitable for our purposes.

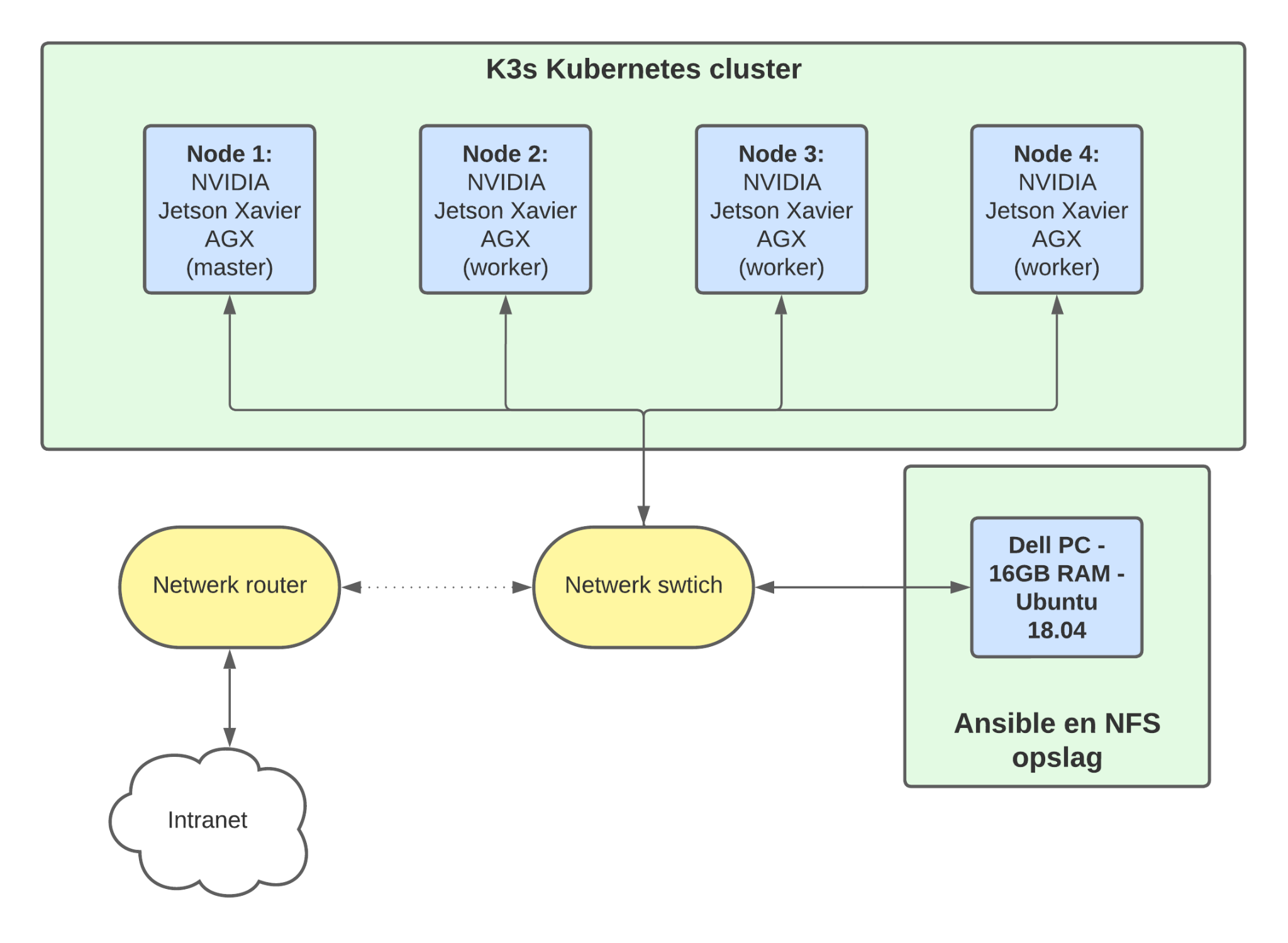
* 1. Low-fidelity Proof of Concept (PoC)

Although the team has access to 4 GPUs, it is not yet possible to effectively run models on them. Therefore, a low-fidelity proof of concept is developed initially, where a quantized version of LLAMA2 is run on a CPU. LangChain is used to provide the model with documents from the SSC-ICT self-service portal as a knowledge base. By experimenting with different parameter settings and prompts, its functionality is optimized. A rudimentary interface is also developed for asking questions.

This phase results in a basic version of our own open-source model. It may not function optimally yet but can be used to validate the idea behind this project and gather initial feedback.

* 1. Local Chatbot

As mentioned earlier, we have access to 4 GPUs. These GPUs are also necessary to run a fully functional LLM. To do this effectively, the GPUs need to be interconnected in a cluster (see fig. 1). Once this is achieved, the models identified as the best options from 1.1 can be tested. This testing can be done in a local environment and will be tailored to the service desk use case. An expected outcome is that models may require fine-tuning before they function effectively. However, this fine-tuning process can be time-consuming and resource-intensive. Therefore, based on this evaluation, one open-source model will be selected for fine-tuning. In this phase, further development will also be carried out on the basic user interface developed in 1.2.



* 1. Fine-tuning the open-source model

In this final phase, we fine-tune the model for our specific purposes. We examine the questions frequently asked at the SSC-ICT service desk and train the model to answer them effectively. This can be done either by providing examples of correct answers or by using reinforcement learning with human feedback (RLHF), where interns from the CFI assess the quality of different responses and further optimize the model based on that feedback.

1. Azure chatbot

For an easier adaptation of AI tools within our organization, we also explore possibilities that can be integrated with existing projects and infrastructure, especially within Microsoft Azure. Azure offers the ability to customize models from OpenAI (e.g., GPT-4) for our own use case. It also provides the option to run this chatbot on-premise. In this case, we can provide a knowledge base, system prompt, and few-shot examples.

To keep the chatbots comparable, we use the same use case as for the open-source option. The 'system prompt' is a description of what the system (the chatbot) is and what it should do. Through trial and error, we will search for a prompt that works well for our use case. Few-shot examples mean that you provide a few examples of how you want it to answer a question, which the system can then extrapolate to other questions. Again, we will use the most frequently asked questions at the SSC-ICT service desk.

The result is a chatbot in the Azure environment that is functionally similar to the chatbot we create based on open-source architecture. This allows us to compare both chatbots and describe the advantages and disadvantages of each approach.

1. Final Report: Comparing Azure Chatbot with Open-Source Model
   1. Establishing and Developing Model Evaluation Criteria

To make a meaningful comparison between an Azure Chatbot and an Open-Source model, it is crucial to first establish the criteria with which we can evaluate the performance of these chatbots. An example could be how effectively they can answer the 100 most frequently asked questions to the SSC-ICT helpdesk. It is important to understand what our stakeholders value in a chatbot. We will attempt to identify important considerations we can quantify, and attempt to measure them. The ultimate result will be presented in the form of a table that shows measurable differences between the two chatbots, such as performance, costs, or energy consumption, to the extent that we can obtain this data.

* 1. Other Differences and Considerations

Comparing an open-source approach with an off-the-shelf product like an Azure chatbot goes beyond quantifiable differences such as performance and energy consumption. There are fundamental differences between these approaches, such as the level of dependence on external parties, flexibility, and the ability to seamlessly integrate them into other services you may use. Our final report will also address these broader aspects of the comparison.

The end product will consist of two chatbots and a report that compares them.