Assignment Submission, WASP Software Engineering Course - Module 2025

Angela Checa
angela.checa@umu.se

August 2025

Introduction

The research area of my interest is the monitoring resources of cloud-edge computing systems, which comprises all the mechanisms to acquire, analyse and store data from a large set of heterogeneous resources such as computation, storage, networks, and applications deployed on cloud-edge infrastructures. The three pillars of monitoring are metrics, logs, and traces. Monitoring is important to gain system insights and helps to understand its performance and health status [1]. The main problem in monitoring the cloud-edge infrastructures is handling massive amounts of data and doing it efficiently and in a lightweight manner, trading the balance between granularity and quantity. Another problem is the bandwidth used to transport the monitored data to a central monitoring system. Since there is too much data that could be useful, other data are redundant and does not provide any additional benefits, and its transmission could be a waste of resources. Finally, a problematic part of monitoring is complying with all requirements in terms of heterogeneity across different layers along the path from the end user, edge devices, fog devices, and the cloud, which involves different provider policies, protocols, and API formats. One initial step in my research is to propose a decentralized multi-agent architecture for monitoring edge-cloud infrastructures with reasoning and filter capabilities about the data collection to avoid redundancies at the same time to act proactively in problem detection.

Lecture principles

The selected concepts are the following:

1. The software engineering "principles"
2. Dynamic techniques for verification and validation

Although my project is in an early stage and the ML concepts are not related with my research topic. I consider the principles of software engineering to be important when building software artifacts to evaluate the models and mechanisms I will propose during my doctoral studies. An adapted version of the software engineering process in which the final consumers of the product are the reviewers and peers could be the key for bringing to the light a solution that could be successfully applied beyond the scope of the academy.

Developing software artifacts to test and evaluate hypothesis need a structure to follow but with enough flexibility to coexist with natural uncertainty of the research process. In that setup I team up with the supervisors then communication and everyday interactions are crucial as within a software company. Agile principles can also be applied to allow everyday progress towards defined milestones of my research.

Likewise, the SE process, the biggest initial effort and most challenging task is to understand the problem. Therefore, requirements engineering can be adopted in this stage as the foundations of an initial version, one that will be refined and tuned through iterations. The

SE principle about the artifact is more important than strict architectural decisions, applies here in the same way considering that during early stages of a research not all the features are decided in advance and the problem may be not completely outlined then it is better to develop artifact's functionalities progressively and always wait to iteration outcomes and new discoveries before put big efforts on creating a long run solution that probably will not lead to anywhere or not provide substantial benefits.

About quality assurance, my plan is to implement periodic reviews of the instruments produced in my project with the help of experts and external research teams. In my opinion, getting fully independent reviews has a high priority because I will depend on different actors to build a testbed of a large-scale system.

With respect to dynamic verification and validation techniques, testing is the concept that is most directly aligned with my research topic. This is because the ultimate goal is to observe the behaviour of heterogeneous systems at massive scale that are naturally complex and need software artifacts to validate wide range of different cases. Manual testing in such environments is not an option, and then automated tests should be carefully designed. The common problems of testing will also be faced, since there are millions of possibilities for applications and configurations that can be monitored in cloud-edge systems, the scope and output space must be set in advance as well as the domains and use cases, which ultimately will drive all the evaluation phases.

Gest-Lecture principles

The guest lecture by Julian Fratiini about requirements engineering is not related to my research in general terms, but the concepts of *Levels of Abstraction* and *System Vision* techniques are the most important takeaway since it could help with the artifacts design process along different phases of my project. I found particularly valuable the insights this lecture gave me about the *levels of abstraction*, for example, in the **Context Layer** my goal will be to recognize: Why should the monitoring system be lightweight? Why should it be decentralized? Why is it important to monitor the continuum of resources in a cloud-edge infrastructure? What are the advantages with respect to other monitoring solutions? In the **Requirements Layer** will be important to resolve the following questions: What are the functions of the monitoring system? and collecting subsequent requirements will be the biggest challenge of my work. Finally, for the **System Layer,** how will be the architecture of the system? how to implement the monitoring of the heterogeneous resources? and how all the interactions of its components will be?

The **system vision** would help me set the scope and limits in which the monitoring system will work. Besides, to establish a global view of all the relations with external entities within the cloud-edge ecosystem and provide understanding about what my solution is going to offer. That vision maybe turns into a tool to manage the iterations and as the entry point to refine the ideas.

Data Scientists versus Software Engineers

- The essential differences between data scientists and software engineers.

I agree with the essential differences highlighted in the book because their roles have distinct but complementary tasks. While data scientists focus only on the ML component and have a model-centric view, engineers prepare the infrastructure and perform tasks throughout the software development cycle with a system-wide view. The former work with model requirements, data cleaning, collection, and labelling; pipelines, algorithms, and model training and evaluation. The latter have a holistic view of the system, and their goal is to deliver a software product under budget and time restrictions. In other words,

data scientists specialize in a part of a larger system that runs ML to enhance its features. In contrast, engineers have a broader action field, they integrate the ML part with all the system components and make possible to run the business around the model, working on different aspects from user requirements, architecture, implementation, testing, deploying at scale until product maintenance.

- Regarding whether those roles will evolve and specialize or will need to learn bidirectionally and whether the roles that will merge.

With the growing trend of data scientist career and considering that data are generated from any area of human life, it is expected to find in that role more people from other backgrounds than software engineering. I hold the opinion that the roles will specialize further, but eventually more T-shape roles will be needed in software development teams or maybe a new intersection role will emerge, a kind of integrator that incorporates the ML part into the system. Similarly, to how the DevOps engineer role emerged.

I believe that nowadays domain-specific data scientists are highly demanded and they have a kind of double I-shaped role (II-shape) because they are experts in data science at the same time have specific knowledge of the business domain the data are related, i.e. financial data scientist. That is why in different industries the need for specialized data scientists is high and in my opinion that trend will not change. The specialization will have double depth of expertise: ML and business level.

Regarding software engineers role, from my perspective, it will become natural for teams to include T-shape members, and I bet for the idea that future successful teamwork guidelines will highlight mandatory to have at least one T-shape member with broad knowledge of ML and expert in software engineering.

Additionally, I expect the T-shape mutation from engineers I-shape side but not from the side of data scientists because I strongly believe that software engineers have advantage over others profiles if they decide to become data scientists, since they have such a background that allows them to understand better the programming skills, mathematics, pipelines, data management and other concepts needed for building a ML component. In contrast to many of today's data scientists from other backgrounds who must learn informatics and computing related topics from scratch.

At this point my opinion differs from the author in [2] regarding *"unicorns"* because in my view it is very likely to find that kind engineer+ML profiles and in the future those *"unicorns"* will turn into *"horses"* that can be easily found in any development team that delivers software products with ML components.

Paper analysis

`I. Engineering LLM Powered Multi-agent Framework for Autonomous CloudOps [3]`

The main driver of the paper goes around that *CloudOps*, that refers to an extension of DevOps practices but for cloud environments, adding to management components like resource discovery, self-healing, and real-time monitoring, this will allow cloud providers to maintain efficient and resilient environments. However, the complex and high dynamic of services and applications on the cloud makes traditional manual operations challenging because they are time consuming and prone to errors. This is why novel solutions involving Generative AI, especially LLMs combined with Retrieval-Augmented Generation (RAG), can be in the scene to demonstrate better performance and accuracy in such tasks.

Main problems to address in CloudOps are managing distributed Data, maintainability, extensibility, and modularity. The most common vendor-specific monolithic approach complicates API integration and makes it difficult to evolve data structures and makes

maintainability a challenging task. MOYA Framework proposed by the authors using multiple agents mitigate those problems because deploy diverse agents as autonomous components designed to manage specific tasks in CloudOps. enabling smart integration with heterogeneous APIs, data sources and workflows at the same time provide flexibility and customization options. Agents can monitor operational activities and generate summaries and specialize in tasks allowing easy maintenance and robustness.

This paper is related to my research because it shares a vision of multi-agent systems specializing in cloud operations including monitoring. Provides an architecture with an important take away that is the agent marketplace idea that found interesting approach to be included in my project.

A fictional project involving LLM and cloud monitoring that my research could fit, is a MOYA framework variation including the capability to answer prompting monitoring queries, in such a way the administrators or SRE can communicate with the monitoring system through a prompt console where they can ask about the status of different services and applications by using natural language. The monitoring system will use dispersed agents communicating with other communities of agents allocated in different system's layers. It could be like plug and play monitoring system that can provide observability functions as result of processing all data collected by agents. For example, in edge layer the SRE could ask how is the performance of the edge servers located in X place and the system will retrieve the texts explaining the dashboards, data, insights, alerts, possible actions and warnings.

A possible tweak of the author's idea of MOYA framework for CloudOps could be to add layer of reasoning agents that connect with multiple cloud providers to create a federation layer then the cloud ops system could be also autonomous in the federation of operations among different clouds. Another option could be to connect the agents to communicate with each other and create a feedback loop to correlate events instead of isolated agents in charge of specific tasks without any connection, an agent that handles the costs could be linked to the agent that provides runtime resources and obtain budget information that allows it to take decisions about resources. Hence, increase the autonomy.

`II. Is your anomaly detector ready for change? Adapting AIOps solutions to the real world [4]`

Concept drift in *AIops* solutions cannot be avoided, but proper model maintenance actions can be taken to overcome the challenge of ever-changing operational data of IT systems. The authors state the importance of drift detectors for monitoring the quality of anomaly detection models. The idea behind it is to alert the system when the model is outdated, then proper model re-training techniques can be applied on time. Another important idea is that different retraining data techniques could be applied depending on the type of anomaly detection techniques used, hence what works for some models could not be the best option for others. These ideas are extremely important because they are a first step toward automated model maintenance pipelines in production that will help data scientists to ensure models quality meaning at the end benefits for overall ML based software product.

This paper connects with the concept of monitoring; in this case, it is not about monitor infrastructure, but rather *concept drift* to keep the anomaly detection models up to date. It added a view of the problems that arise from the extreme data dependency of *Alops* solutions, which are to some extent related to cloud monitoring systems. In addition, anomaly detection models are currently being used to monitor cloud-based solutions to increase proactivity [5]. Considering the vast amount of data generated by monitoring

tools, an important topic for my research could be scaling telemetry mechanisms to reduce the amount of data collected at the same time be able to deliver details about events and system status with enough granularity. From the paper I consider valuable the way they define the metrics for evaluating the precision and performance of anomaly detectors.

An AI-intensive project or that enthralled me recently is Zero Touch Operations (ZTO), specifically in the context of networks. With ZTO, humans will only provide high-level business goals to the system and will translate into low-level rules and configurations, assisted with ML algorithms, the network will be optimized and adapted to changes in an autonomous way [6]. Here anomaly detection in networking and prediction of KPI could play a key role in reducing the OPEX of communication service providers (CSP) [7].

In Zero touch networks and management operations (ZSM), the concept of automated ML (AutoML) emerge as a key tool for automating the ML pipelines and improve the efficiency of the solutions [8], in consequence concept drift is also a concern, and for the automatized model updating tasks the idea presented in the project fits very well but instead of focus on the CPU utilization and internet traffic, it could be good approach to reuse those ideas to analyse the KPI of 5G networks. Depending on the type of 5G deployment scenarios like eMBB, mMTC and uRLLC, a high level of ML Models could run different drift detection strategies. On the other hand, as KPI of each scenario commands separate groups of resources in cascade, the highest KPI based ML model detector could help to efficiently update the cascade down of other ML models running on different parts of the network that involves each scenario.

My WASP research could fit into ZTO in the sense of making monitoring tools more intelligent give feedback instead of producing metrics and dashboards to the system and automatize the operations needed to reestablish the status of the system after an event detected by monitoring system. I might tweak my research towards anomaly detection on 6G systems for ZTO, by adapting the paper ideas for drift-based anomaly detection models, and apply them to *eMBB*, analysing the advantages of those detectors in each service layer, and a goal could be to identify anomalous patterns of speed, latency and bandwidth that affect KPI of different virtual layers.

Research Ethics & Synthesis Reflection

The paper search process was as follows: - visit latest CAIN conference web page and check the section of published papers. From the list of long papers I scanned the titles in search for those with the words *Cloud* OR *Monitoring* OR *Agents*, from that the only one chosen was the reference [3] because it contains the word *CloudOps* and *agent*, proceed to find the document using google scholar, finally read the abstract, the keywords and the introduction, searching for relations with monitoring, when found decided that the paper match with my interest and could be useful. Since the first title scanning was not enough, I read the abstract of all the papers with the same criteria.

At the beginning when scanning the titles, I found daunting that most of the topics are about neural networks, LLM or RAG, feeling that my research does not fit in those areas and for my area does not have relevance. My first approach was discarding those with deep topics of LLM and IA. Although, when found the papers I realized that there are options to apply ML in cloud systems that could potentially help to my project view. In the last conference (2025) found paper [9] as an option because of analyse metrics of devices energy usage but did not choose because it is not directly applicable. related with monitoring systems.

As no other papers from last conference fit with my search formula I moved to the website of CAIN 2024 and accomplish the same process, this time no title contain the word *Cloud*. However, thanks to the reading done in section 4 of this assignment, I learned that *AIOps* is the are ML systems that learn from operational data of IT systems, then

identify that paper [4] was an alternative, when read the keywords and abstract realize that is related with model monitoring and contain ideas related with failure prediction and anomaly detection which are topics linked to monitoring systems.

Regarding ethics, I avoid the use of AI chatbots to generate or refine my texts, the sentences are completely original and come from the ideas that came up after reading the course book, papers and other literature found during the process of completing this assignment.

References

[1] V. K. Sikha, "The SRE Playbook: Multi-Cloud Observability, Security, and Automation," *Journal of Artificial Intelligence & Cloud Computing*, vol. 2023, no. September, pp. 1–7, 2023.

[2] C. Kästner, *Machine Learning in Production: From Models to Products*. Cambridge, MA: The MIT Press, 4 2025.

[3] K. Parthasarathy, K. Vaidhyanathan, R. Dhar, V. Krishnamachari, A. Kakran, S. Akshathala, S. Arun, A. Karan, B. Muhammed, S. Dubey, and M. Veerubhotla, "Engineering llm powered multi-agent framework for autonomous cloudops," in *2025 IEEE/ACM 4th International Conference on AI Engineering – Software Engineering for AI (CAIN)*, pp. 201–211, 2025.

[4] L. Poenaru-Olaru, N. Karpova, L. Cruz, J. S. Rellermeyer, and A. van Deursen, "Is your anomaly detector ready for change? adapting aiops solutions to the real world," in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*, CAIN '24, (New York, NY, USA), p. 222–233, Association for Computing Machinery, 2024.

[5] C. K. Huang and G. Pierre, "Acala: Aggregate monitoring for geo-distributed cluster federations," in *Proceedings of the ACM Symposium on Applied Computing*, pp. 156–164, Association for Computing Machinery, 6 2023.

[6] J. Crawshaw, "Lessons in Zero-Touch Operations from the Coalface," tech. rep., OMDIA & Ericsson Report, 2022.

[7] E. Alberti, S. Alvarez-Napagao, V. Anaya, M. Barroso, C. Barrué, C. Beecks, L. Bergamasco, S. A. Chala, V. Gimenez-Abalos, A. Graß, D. Hinjos, M. Holtkemper, N. Jakubiak, A. Nizamis, E. Pristeri, M. Sànchez-Marrè, G. Schlake, J. Scholz, G. Scivoletto, and S. Walter, "Ai lifecycle zero-touch orchestration within the edge-to-cloud continuum for industry 5.0," *Systems 2024, Vol. 12, Page 48*, vol. 12, p. 48, 2 2024.

[8] M. El Rajab, L. Yang, and A. Shami, "Zero-touch networks: Towards next-generation network automation," *Computer Networks*, vol. 243, p. 110294, Apr. 2024. arXiv: 2312.04159 Publisher: Elsevier.

[9] V. Nguyen, V. Dhopate, H. Huynh, H. Bouhlal, A. Annengala, G. L. Scoccia, M. Martinez, V. Stoico, and I. Malavolta, "On-device or remote? on the energy efficiency of fetching llm-generated content," in *2025 IEEE/ACM 4th International Conference on AI Engineering – Software Engineering for AI (CAIN)*, pp. 72–82, 2025.