A blue bird with a white background

Description automatically generated

**7030CEM – Project Proposal**

**NAME: CELESTINA CLAREMONT**

**STUDENT ID: 14440216**

**Table of Contents**

[Section A – Ethics Application 3](#_Toc180005391)

[Section B – Project Proposal 3](#_Toc180005392)

[I. Research Question/Problem Statement 3](#_Toc180005393)

[II. Intended Users and Their Requirements 5](#_Toc180005394)

[III. System Requirements, Project Deliverables, and Final Project Outcome 5](#_Toc180005395)

[IV. Primary Research Plan 6](#_Toc180005396)

[V. Initial Literature Review 7](#_Toc180005397)

[VI. Bibliography 10](#_Toc180005398)

[VII. Appendix 12](#_Toc180005399)

# Section A – Ethics Application

|  |  |
| --- | --- |
|  | I submitted my ethics application, and my application has been approved. I include my ethics certificate in the appendix as evidence. |
|  | I submitted my ethics application, and my application is currently under review. |
|  | I have not submitted my ethics application. |

# Section B – Project Proposal

### Research Question/Problem Statement

The central research question is: How can federated learning be applied to detect API rate-limiting violations in Kubernetes-based microservice architectures using pre-trained anomaly detection models while ensuring privacy and scalability? This research investigates how federated learning (FL) can improve API security by detecting misuse in distributed microservice environments, addressing privacy concerns, and enhancing scalability.

#### Problem Statement:

In modern applications, APIs enable communication between microservices *(Singh et al., 2020)*. However, with the rise of API-based architectures, API rate-limiting violations, where clients exceed the allowable number of requests, have become a growing concern *(Krieger et al., n.d.,)*. These violations can lead to severe performance degradation, unauthorised access, or denial of service (DoS) attacks *(Serbout et al., 2023)*.

This dissertation tailors federated learning to detect rate-limiting abuses, which are increasingly critical in cloud-native architectures. The novelty of this dissertation lies in its focus on using federated learning to address API rate-limiting within microservices. Centralised anomaly detection techniques struggle with scalability and data privacy issues, particularly in distributed environments, where aggregating API data can breach privacy regulations and degrade system performance (*Lu et al., 2023*). Thus, a decentralised and privacy-preserving solution is required to address the growing threat of API abuse in microservice environments.

#### Evidence of the Problem:

Empirical research highlights the increasing prevalence of API abuse, particularly focusing on rate-limiting violations. El Malki et al. [(2022) discuss the growing risk posed by API overuse and abuse in microservices, pointing out the performance degradation and security vulnerabilities that arise when APIs are abused without proper rate-limiting controls](https://ieeexplore.ieee.org/abstract/document/9912639). They argue that these abuses can significantly impact the reliability of microservices, leading to reduced user satisfaction and costly downtime.

Serbout et al. (2023) emphasise the importance of machine learning-based anomaly detection for securing API services in real time. [Their study demonstrated that machine learning models effectively detect anomalies in API traffic patterns, including rate-limiting violations, enhancing both efficiency and security in microservice environments](https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-023-00471-1). These studies prove that API rate-limiting violations are a real and pressing issue in modern cloud-based architectures.

#### Approach:

To address the problem of detecting API rate-limiting violations in Kubernetes-based microservice architectures, this dissertation will combine federated learning (FL) and pre-trained anomaly detection models. Federated learning allows anomaly detection models to be trained locally on individual microservices without sharing raw API data, thus preserving privacy *(Vucovich et al., 2022)*. Each microservice will train a local model based on API request patterns and send aggregated model updates (rather than raw data) to a central server *(Vucovich et al., 2022)*. This ensures the system has a global understanding of API activity while protecting sensitive data.

Using pre-trained models will further reduce the complexity and time required for model development. These models, specifically tuned for detecting API rate-limiting violations, will be fine-tuned within the federated learning framework to identify abnormal API usage patterns. The system will be implemented in a Kubernetes cluster to test federated learning’s ability to detect API abuse in microservice environments.

### Intended Users and Their Requirements

There is a growing need for this dissertation due to the increasing reliance on APIs for communication in microservice architectures *(Wang et al., 2021)*. As organisations adopt cloud-native applications, the risk of API rate-limiting violations becomes more pronounced. These violations can lead to significant performance issues, unauthorised access, and denial of service (DoS) attacks, which can compromise the reliability and security of microservices *(Atlassian, n.d.,)*. Centralised anomaly detection techniques struggle with scalability and data privacy issues, particularly in distributed environments such as Kubernetes, where aggregating API data can breach privacy regulations and degrade system performance (*Lu et al., 2023*). Therefore, a decentralised and privacy-preserving approach, such as federated learning, is essential to effectively detect and mitigate API rate-limiting violations. The primary users of this research are

1. **Developers and engineers** working in cloud-native environments, particularly those involved in managing Kubernetes-based microservice architectures.
2. **Cybersecurity professionals and researchers** focusing on API security and anomaly detection will benefit from the findings.
3. **Academic researchers** studying federated learning and its applications in distributed systems are also key beneficiaries.

### System Requirements, Project Deliverables, and Final Project Outcome

#### System Requirements

* Federated Learning Framework tools for distributed model training.
* Anomaly Detection: Pre-trained models to detect API rate-limiting violations.
* API Monitoring: Prometheus for tracking API traffic.
* Kubernetes: For microservice orchestration and deployment.
* Cloud-based deployment of microservices and resources.

#### Project Stages and Deliverables:

The dissertation will proceed through the following stages:

1. **Kubernetes Setup (Week 1-2)**:
   * Set up the Kubernetes cluster for deploying the microservices.
   * **Deliverable**: Functional Kubernetes cluster with microservice deployment.
2. **API Rate-Limiting Simulation (Week 2-3)**:
   * Implement rate-limiting features and simulate API violations.
   * **Deliverable**: API environment for testing violations.
3. **Federated Learning Implementation (Week 3-4)**:
   * Deploy federated learning to detect rate-limiting violations in microservices.
   * **Deliverable**: Fully functioning federated learning framework across the microservices.
4. **Testing and Evaluation (Week 4-5)**:
   * Test the system for scalability, privacy, and accuracy in detecting API abuse.

#### Final Project Outcome:

The final product will employ federated learning to ensure data privacy while monitoring API traffic. The final deliverable will include:

* A comprehensive report on system testing results, highlighting its effectiveness, privacy benefits, and scalability.
* Prototype implementation demonstrating federated learning in a Kubernetes environment to detect API abuse in microservices.

### Primary Research Plan

The following primary research methods will be employed:

1. **Simulation and Experimentation**
   1. **API Rate-Limiting Violations:** A simulation of API requests in a Kubernetes-based microservice environment will be created to generate data on rate-limiting violations.
   2. **Federated Learning Implementation:** Federated learning will be applied across multiple microservices to detect API anomalies. Each microservice will train a local model using its API request data and send updates to a central server for aggregation. The pre-trained anomaly detection models will identify API abuse patterns, focusing on rate-limiting violations.
2. **Data Collection and Analysis**
   1. **Data Collection:** Cloud Storage will collect and store API traffic logs. The Cloud Logging service will be used to capture API usage over time.
   2. **Amount of Data:** API traffic logs will be collected over a simulated period of 1-2 weeks, with API requests generated at different rates to simulate violations.
   3. **Analysis:** The system’s ability to detect anomalies will be evaluated. The analysis will focus on how federated learning can train models across microservices without sharing raw data.
3. **System Testing** 
   1. **Testing Criteria**: Tests will assess the system’s performance in the detection of rate-limiting violations.
4. **Sequence of Tasks:**
   * Set up Kubernetes environment and implement basic API monitoring.
   * Simulate API rate-limiting violations.
   * Implement federated learning and begin data collection through simulations.
   * Analyze collected data and refine the system.
   * Final testing, performance analysis, and report generation.

### Initial Literature Review

The paper "Anomaly Detection Using Federated Learning" by Shubham Singh, Shantanu Bhardwaj, Hemlatha Pandey, and Gunjan Beniwal explores the use of federated learning to detect anomalies in data streams without centralising the data. They implemented an autoencoder neural network model using PyTorch and PySyft across two clients to perform anomaly detection. Their results indicated that the federated model outperformed models trained on the full dataset. Their work aligns with my dissertation, as both focus on detecting anomalies. However, while this paper centres on anomaly detection in general datasets, my dissertation focuses on using federated learning to detect API abuse in Kubernetes-based microservices, addressing a more niche aspect of anomaly detection in modern cloud architectures.

The paper "Multi-task Federated Learning-based System Anomaly Detection and Multi-Classification for Microservices Architecture" by Junfeng Hao et al. introduces the SADMC-MT-FF-FL framework, which applies multi-task federated learning to detect and classify anomalies in microservices architectures. The framework utilises feature fusion and parallel knowledge transfer to improve anomaly detection across distributed microservices, achieving superior accuracy in experiments on benchmark platforms. However, challenges like communication overhead, data privacy, and non-IID data distribution remain. This research aligns closely with mine as it addresses anomaly detection in distributed microservices using federated learning. While their approach focuses on multi-class classification of anomalies, my dissertation will focus on API rate-limiting violation detection, providing insights into a more niche application of federated learning in cloud-native architectures.

The paper "Impact of API Rate Limit on Reliability of Microservices-Based Architectures" by Amine El Malki, Uwe Zdun, and Cesare Pautasso investigates the effect of API rate-limiting on the reliability of microservices. Using an analytical model, the authors tested different workload configurations on private clouds and Google Cloud Platform (GCP) to predict success and failure rates. The model demonstrated low error rates in prediction, proving its potential in optimising rate limits for system reliability. However, further research is needed to explore different architectures and configurations for generalisation. This research closely aligns with my dissertation, as it highlights the importance of rate-limiting in maintaining system reliability.

The paper "API Rate Limit Adoption – A Pattern Collection" by Souhaila Serbout et al. explores the various patterns associated with the adoption of API rate limiting. It emphasises the importance of controlling API request rates to prevent abuse and ensure fair resource allocation. The authors categorise these patterns into seven sub-collections based on specific challenges, such as setting rate limits, metering usage, and reacting to limit violations. This research is closely related as it provides essential insights into API rate-limiting patterns that can inform the design of a federated learning system for detecting rate-limiting violations in microservices.

The paper "FADngs: Federated Learning for Anomaly Detection" by Boyu Dong et al. addresses the challenges of applying federated learning (FL) to anomaly detection, emphasising the need for privacy-preserving techniques. The proposed method, FADngs, uses noisy global density estimation and self-supervised ensemble distillation to improve anomaly detection by aligning data distributions across clients and enhancing model representations. This research is relevant as it explores advanced federated learning techniques specifically tailored for anomaly detection, complementing my focus on detecting API rate-limiting violations.

### Bibliography

1. El Malki, A., Zdun, U., & Pautasso, C. (2022). Impact of API Rate Limit on Reliability of Microservices-Based Architectures. In *IEEE Conference Publication | IEEE Xplore*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9912639
2. *FADngs: Federated Learning for Anomaly Detection*. (n.d.). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10409269
3. Hao, J., Chen, P., Chen, J., & Li, X. (2024). Multi-task federated learning-based system anomaly detection and multi-classification for microservices architecture. *Future Generation Computer Systems*, *159*, 77–90. https://doi.org/10.1016/j.future.2024.05.006
4. Krieger, B., TestifySec, Ericsson Software Technology, Kautz, F., Bork, J., NYU, Pruchniak, M., VMware, Faisal, R., Red Hat, Krieger, B., Kennedy, C., Değirmenci, F., Kautz, F., Bork, J., Moore, M., Pruchniak, M., Joglekar, P., Faisal, R., . . . Raghunathan, S. (n.d.). *Cloud native security whitepaper*. https://www.cncf.io/wp-content/uploads/2022/06/CNCF\_cloud-native-security-whitepaper-May2022-v2.pdf
5. Lu, T., Wang, L., & Zhao, X. (2023). *Review of Anomaly Detection Algorithms for Data Streams*. *Applied Sciences*, *13*(10), 6353. https://doi.org/10.3390/app13106353
6. Serbout, S., Malki, A. E., Pautasso, C., & Zdun, U. (2023). API Rate Limit Adoption -- A pattern collection. *EuroPLoP ’23: Proceedings of the 28th European Conference on Pattern Languages of Programs*. https://doi.org/10.1145/3628034.3628039
7. Singh, S., Bhardwaj, S., Pandey, H., & Beniwal, G. (2020). Anomaly Detection Using Federated Learning. In *Advances in intelligent systems and computing* (pp. 141–148). https://doi.org/10.1007/978-981-15-4992-2\_14
8. Vucovich, M., Tarcar, A., Rebelo, P., Gade, N., Porwal, R., Rahman, A., Redino, C., Choi, K., Nandakumar, D., Schiller, R., Bowen, E., West, A., Bhattacharya, S., & Veeramani, B. (2022, October 12). *Anomaly Detection via Federated Learning*. arXiv.org. <https://arxiv.org/abs/2210.06614>
9. Wang, Y., Kadiyala, H., & Rubin, J. (2021). Promises and challenges of microservices: an exploratory study. *Empirical Software Engineering*, *26*(4). https://doi.org/10.1007/s10664-020-09

### Appendix

#### Ethics Approval Confirmation Letter

A certificate of ethical approval

Description automatically generated