MyMONIT

Collecting measurements to monitor CERN’s experiments

Table of Contents

Word count: 1.012

High-level description

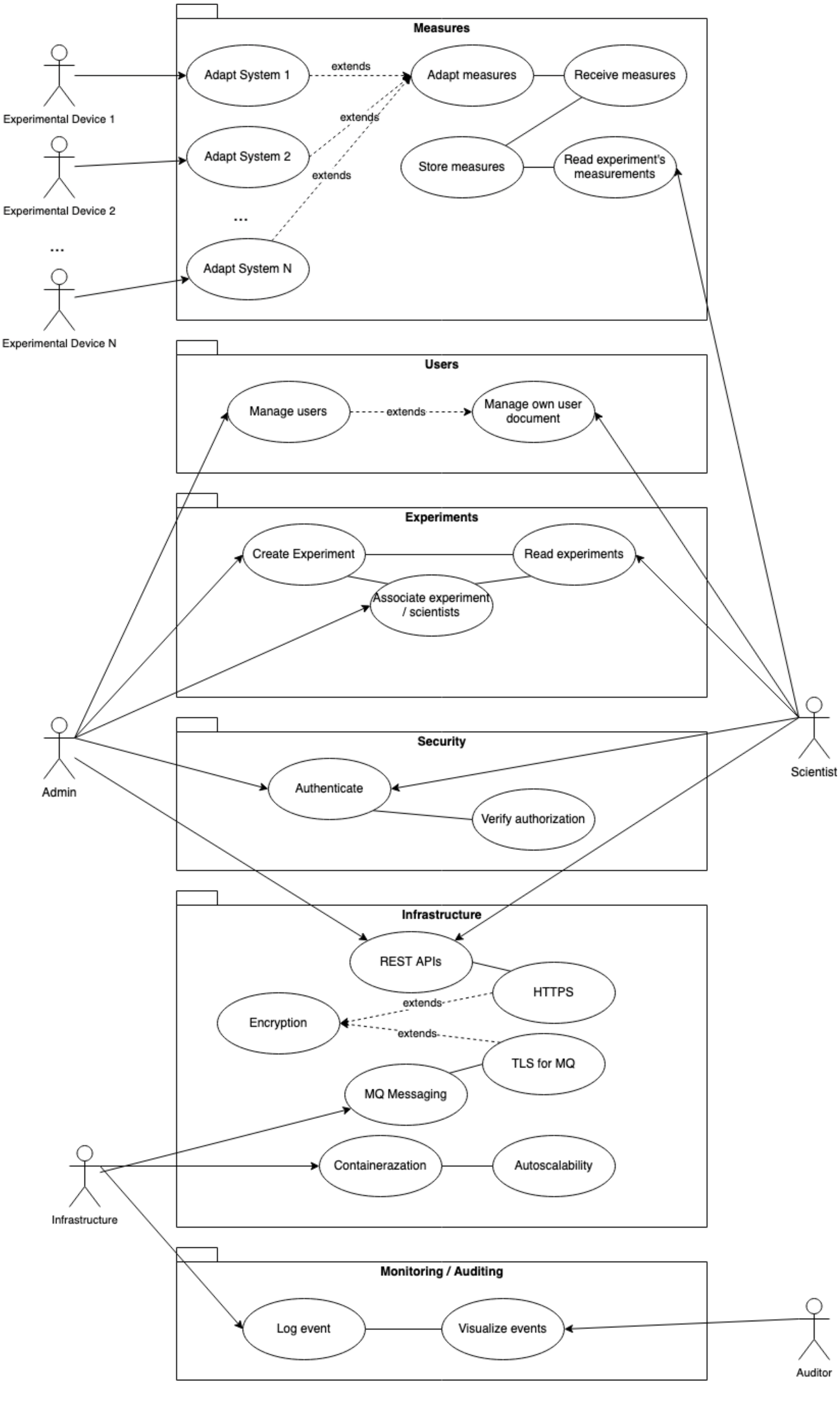
CERN uses a variety of independently developed systems to monitor its infrastructure (Aimar et al., 2019). MyMONIT will be a solution to unify the monitoring of experiments into a single software integrating different streams of measurements to centralize this information.

MyMONIT will be scalable to ensure that it can cope with increasing demand. The solution will also include auditing to detect anomalies in the system itself and the flow of the measurements.

MyMONIT will store confidential information and will be a key component in the monitoring infrastructure. Adequate security measures will be implemented to address the associated risks.

Requirements

The following diagram illustrates all the use cases.



Functional requirements

- There will be three user types with the following role matrix:

role	resource	scope	access
Administrators	users	complete	RW
	experiments	complete	RW
	measurements	complete	R
	audits	no access	/
Scientists	users	only user's record	RW
	experiments	only records associated with the user	R
	measurements	only records associated with user's experiments	R
	audits	no access	/
Auditors	users	no access	/
	experiments	no access	/
	measurements	no access	/
	audits	complete	R

- For each source, an adapter will normalize the measure and transmit it to MyMONIT.
- The measures will be persisted, indexed per experiment, and made available through APIs to authorized scientists.
- A complete audit will be available from a separate interface.

Non-Functional Requirements

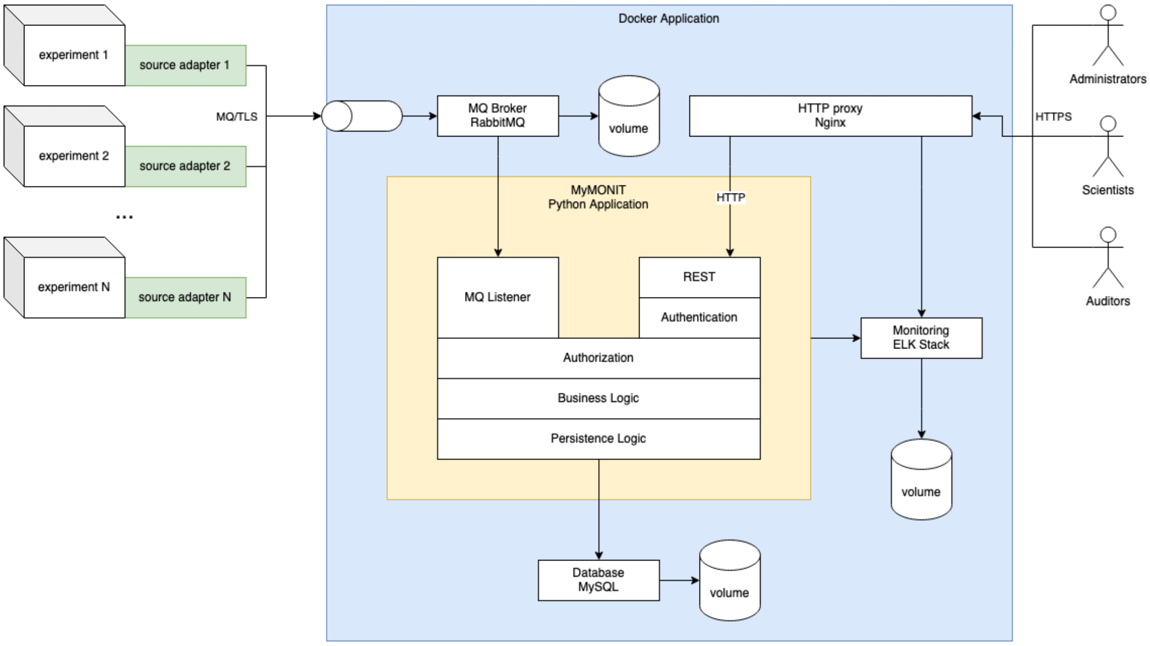
- Access points will be authenticated.
- Access to experiments will require per-user authorization.

- The system must serve concurrent users and concurrent experiments.
- 100% of data must be retained.
- The attack surface must be limited.
- Automated testing and code scanners will support maintainability.

Assumptions

- The system's capacity must accommodate at least 10 years of data.
- Autoscale functionalities will be sufficient to deal with variable demand (Kubernetes, N.D. a).
- It is expected an elevated flow of measurements and that queues will absorb peaks of traffic (Reagan, 2018).
- Users will visualize measures polling the APIs and pagination will be sufficient to reduce the performance load.
- The total number of users will be in the range of a few thousand.
- Experiments will produce less than 1 million measurements each.

Architecture



The adapters (in green) will send the measurements to the solution (in blue) where the main component (in yellow) will index them and expose them via REST APIs.

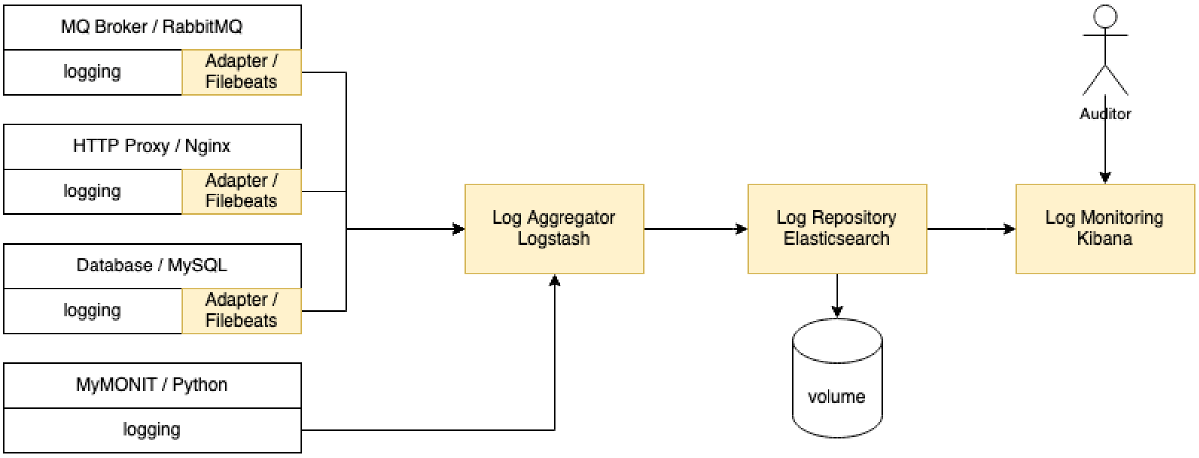
- Docker and Docker Compose: the solution will be containerized and will be portable to compatible solutions such as Kubernetes (Kubernetes, N.D. b).
- The adapters will be Python scripts customized for each specific case and will be installed at the experiment’s location.

- Nginx will be used as a reverse proxy with SSL offloading and will hide all HTTP resources from the outside network. Nginx is currently one of the market leaders in this field (W3Techs, 2022).

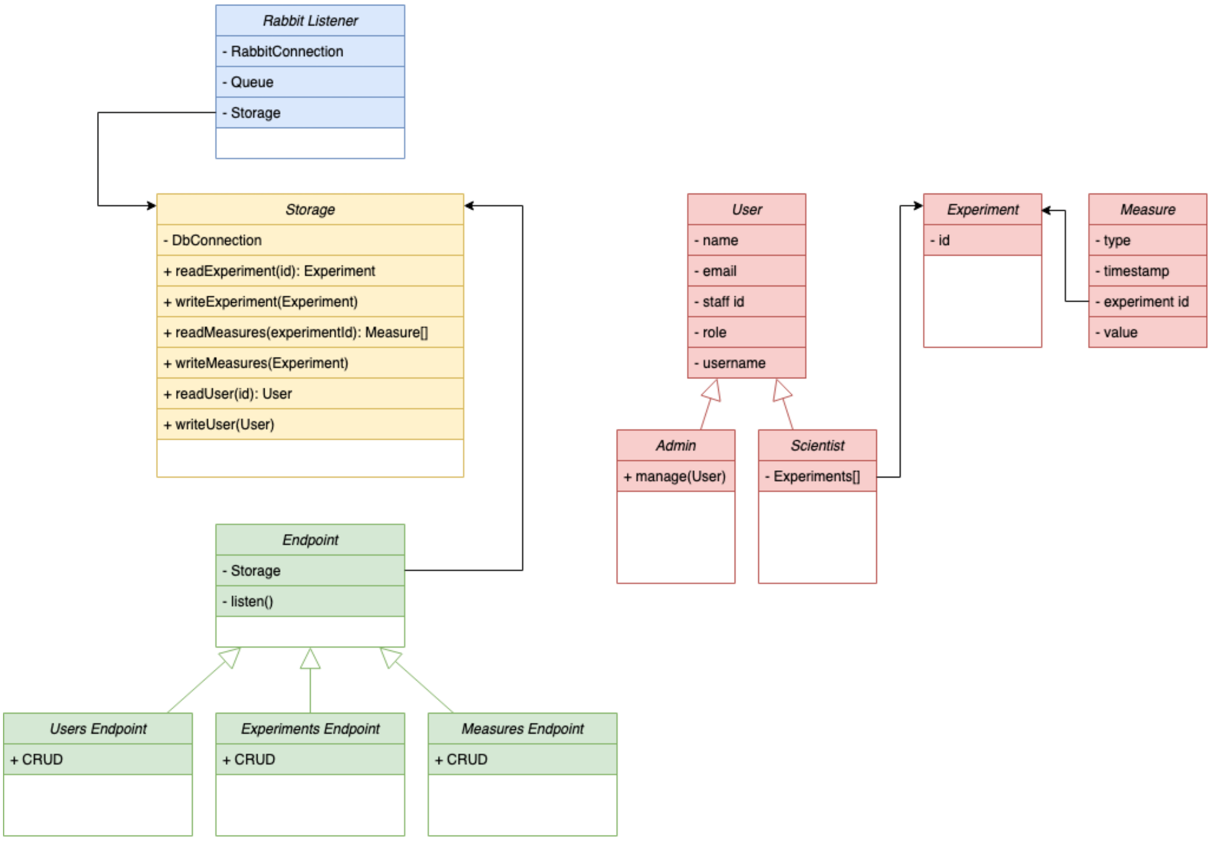
- RabbitMQ will be used as MQ Broker to accept encrypted data streams from the experiments. RabbitMQ is a popular solution and it was preferred to Kafka because it guarantees global message ordering in a cluster (Souza, 2020) even if Kafka offers better scalability for high volumes of traffic (Rabiee, 2018; Souza, 2020)

- MySQL will be responsible for the storage of the application's data. The design will allow to replace it with a more scalable NoSQL database if necessary (Khasawneh, 2020).

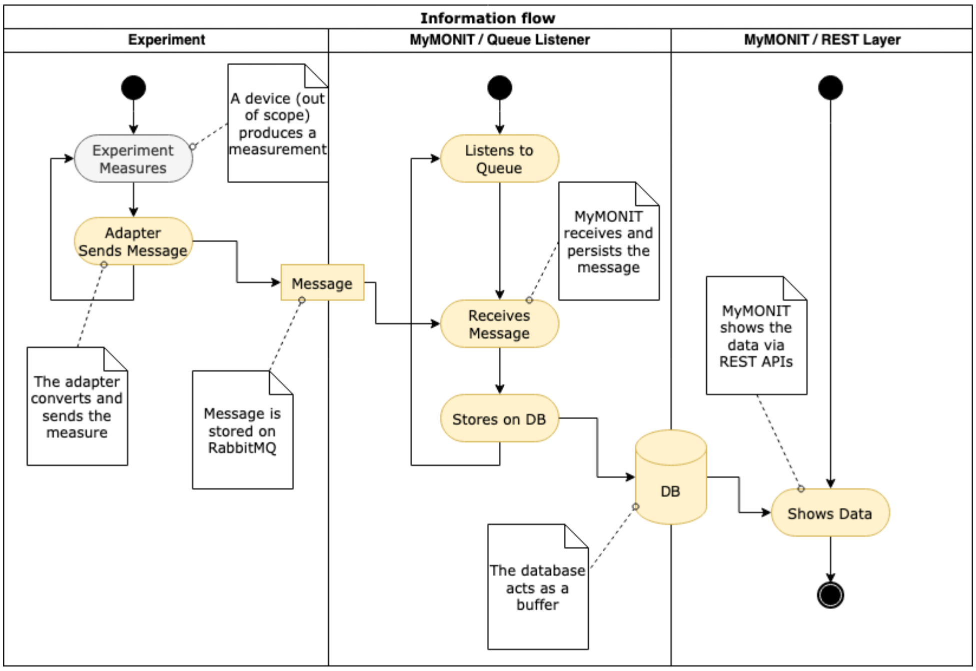
- ELK Stack (Elastic Search, Logstash, and Kibana) will be used for log collection and dashboarding. Filebeats will be used as an adapter where needed. ELK Stack is the only open source among the most popular solutions of this kind (Gillespie & Givre, 2021).



- MyMONIT will be a Python application using Flask and Pika. Flask allows for rapid web development (Ghimire, 2020). Pika is the recommended library to support RabbitMQ in Python (RabbitMQ, N.D.). The following diagram illustrates the internal design of MyMONIT. There will be no direct interactions between the components consuming messages from the broker (in blue) and the components exposing REST endpoints (in green). The Storage (in yellow) will mediate the communications between the two parts.

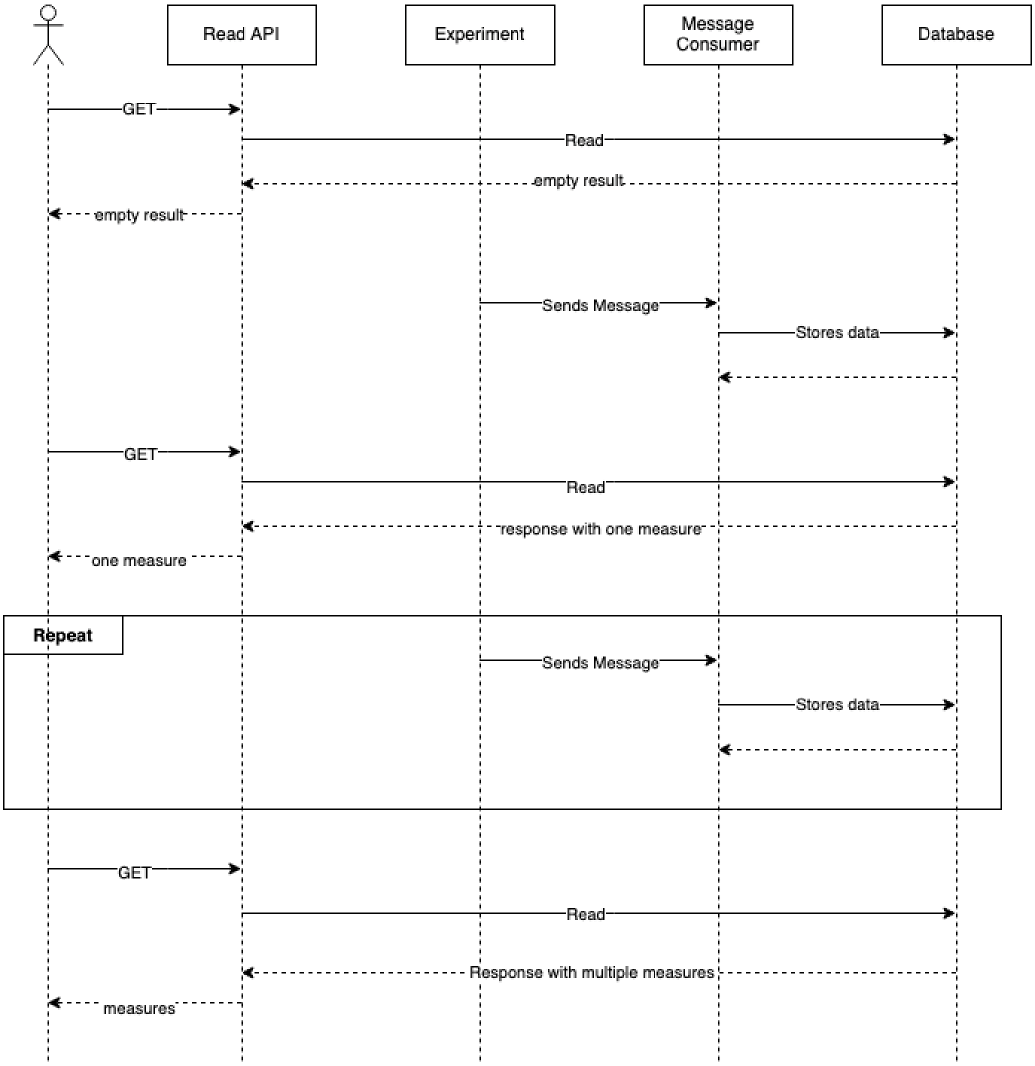


Information flow


Information flow
Experiment
MyMONIT / Queue Listener
MyMONIT / REST Layer
A device (out of scope) produces a measurement
Experiment Measures
Adapter Sends Message
Message
The adapter converts and sends the measure
Message is stored on RabbitMQ
Listens to Queue
MyMONIT receives and persists the message
Receives Message
Stores on DB
DB
The database acts as a buffer
MyMONIT shows the data via REST APIs
Shows Data

The diagram shows from a location perspective how information flows between components.

The following diagram, instead, represents the same flow from a time perspective:



Security

Overview

The main security concerns are the risks of sabotage and information leak. Being a monitoring tool, an attacker may try to disrupt the operations to cover another attack. Information leaks could endanger the process of peer reviews allowing scientists to steal data from parallel research. Being an application exposed only to an internal network, cyberattacks from external sources will be limited.

Authentication

The authentication will be based on JSON web tokens that will remain valid for a limited time and will be required in all interactions. A shared secret (API key) will also be required to limit the chances of brute force attacks (OWASP, REST Security).

Authorization

Authorization to the users, experiments, and measurements endpoints will be role-based. Auditors will have full access limited to audits.

Code quality

- Secure coding practices (OWASP, Secure coding)
- Automated code scanners

Auditing

The main goal will be:

- identification of incidents and fraudulent activity
- detection of anomalies

The following events will be logged:

- failed authentications
- authorization failures
- throughput

The following data will never be logged:

- credentials and tokens
- personal data, except for staff identification

(OWASP, Logging)

Security Risks

Using the STRIDE model, the following threats were identified and classified with DREAD (OWASP, Threat Modeling).

Spoofing

User's credentials violation	
Type	Level
Damage	High (10), experiments would be exposed, users' records compromised, data leak
Reproducibility	High (10)
Exploitability	High (10)
Affected users	Low-Medium (4). One user. All, if the user is an administrator
Discoverability	Medium (6). User's credentials may be easy to guess
DREAD	High (8)
Mitigation	Password policy: minimum complexity with expiration

Measurements adapter’s credential violation	
Type	Level
Damage	High (10). It could allow for DDoS on queues or tampering

Reproducibility	Low (2). The audit will reveal additional login attempts
Exploitability	High (10). If discovered, credentials could be easily used to authenticate scripts
Affected users	Low (2). One experiment
Discoverability	Medium (6). Adapters may be poorly designed with low security in their design
DREAD	Medium (6)
Mitigation	Complex passwords with password rotation

Cross-site request forgery	
Type	Level
Damage	High (10). Administrators may accidentally modify data
Reproducibility	Low (1). It would be very difficult to perform such an attack
Exploitability	Medium (5). The setup may be easy
Affected users	High (8). Potentially all users
Discoverability	Low (1). The attacker needs a deep understanding of the system
DREAD	Medium (5)
Mitigation	Correct APIs design, usage of a token (OWASP, CSRF)

Tampering

An employee installs tampered measurement adapter	
Type	Level
Damage	High (10), experiments would be invalidated
Reproducibility	Medium (6). The highest risk is the broker's authentication
Exploitability	High (10). Employees in certain positions have easy access
Affected users	High (10). All scientists
Discoverability	Medium (6). Employees in certain positions have easy access
DREAD	High (8.4)
Mitigation	Mandatory lifecycle management for production software, including measurement adapters, with mandatory peer approval to promote software to production

Employee manipulates audits

Type	Level
Damage	Medium (5), it could be part of a more vast attack and it could delay the detection of an issue
Reproducibility	Low (1). It requires another violation
Exploitability	Low (1). It is hard to manipulate audits stored in Elasticsearch
Affected users	Low (3). Auditors
Discoverability	Low (1). Elasticsearch is not directly exposed. Only a limited number of employees could easily explore possible attacks.
DREAD	Low (2.2)
Mitigation	Access to the filesystem must be restricted. The filesystem should be encrypted.

Administrator manipulates documents

Type	Level
Damage	Medium (4), data could be recovered through backups, activities could suffer delays
Reproducibility	High (10). Administrators could easily manipulate records
Exploitability	High (10). Administrators can manipulate records as part of their role
Affected users	High (10). All scientists
Discoverability	High (10). Administrators can manipulate records as part of their role
DREAD	High 8.8
Mitigation	Monitoring and auditing will detect fraudulent activity. Screening of employees in this role is recommended.

Repudiation

The User denies committing an action

Type	Level
Damage	Low (1).
Reproducibility	Low (1). All actions are audited. Administrators do not have W access to audits
Exploitability	Low (1). Administrators do not have W access to audits

Affected users	Low (1).
Discoverability	Low (1). Without an attack on audits, repudiation would be ineffective
DREAD	Low (1)
Mitigation	No mitigation is necessary

Information disclosure

Database breach	
Type	Level
Damage	High (10), data would be compromised.
Reproducibility	Low (1). The database is not directly exposed, authentication is in place
Exploitability	Low (1). The attacker should compromise at least another system first
Affected users	High (10). All
Discoverability	Low (1). Only a few employees could easily explore attacks
DREAD	Medium (4.6)
Mitigation	The database won't be exposed to the external network, access will be authenticated

Scientists stealing information	
Type	Level
Damage	Medium (5). Peer reviews may be invalid
Reproducibility	Low (1). It requires another violation
Exploitability	Low (1). It requires another violation
Affected users	Medium (5). Scientists involved in the experiments, external stakeholders
Discoverability	Low (1)
DREAD	Low (2.4)
Mitigation	No mitigation will be implemented. The employer's disciplinary procedures should be a sufficient deterrent.

Auditors steal information through audits	
Type	Level

Damage	Medium (4). Peer reviews may be invalid. Security may be compromised
Reproducibility	High (10). Auditors have access to audits as part of their role
Exploitability	High (10). Auditors have access to audits as part of their role
Affected users	High (10). Administrators, Scientists, and Stakeholders
Discoverability	High (10). Auditors have access to audits as part of their role
DREAD	High (8.8)
Mitigation	Auditors’ actions will be audited as well. The employer’s disciplinary procedures should be a sufficient deterrent.

Denial of service

DDoS on APIs	
Type	Level
Damage	High (10). The system may become inoperative
Reproducibility	Low (3). The system should be exposed only in the internal network
Exploitability	Low (3). It would be easy to block the attack in the internal network
Affected users	High (10). All
Discoverability	Low (1). It would be difficult to plan an effective attack.
DREAD	Medium (5.4)
Mitigation	Out of scope in this project. The system administrator must be able to isolate the segment of the network causing the attack.

DDoS on Audit and Monitoring	
Type	Level
Damage	Medium (6). It may cover a more vast attack
Reproducibility	Low (1). The system should be exposed only in the internal network
Exploitability	Low (1). It would be easy to block the attack in the internal network
Affected users	Low (3). Auditors
Discoverability	Low (1). It would be difficult to plan an effective attack.
DREAD	Low (2.4)

Mitigation	Out of scope in this project. The system administrator must be able to isolate the segment of the network causing the attack.

Ransomware attack	
Type	Level
Damage	High (10), all data may be lost
Reproducibility	Medium (6). Measures are in place, but everyday organizations fall under this attack
Exploitability	High (10). The attack may come in the form of phishing.
Affected users	High (10). All
Discoverability	Medium (6). It is hard to evaluate the level of the current defenses
DREAD	High (8.4)
Mitigation	MyMONIT's network will be in a separate segment and virtualized in the container infrastructure. Containers' images will be maintained up to date. Offline backups will ensure the recoverability of data. (OWASP, Ransomware)

Elevation of privilege

Scientists becoming administrators	
Type	Level
Damage	High (8), the attacker could disrupt the system
Reproducibility	Low (1). It would require database access since no system function manipulates roles
Exploitability	Low (1). The attacker should compromise at least another system first
Affected users	High (10). All
Discoverability	Low (1)
DREAD	Medium (4.2)
Mitigation	Code reviews and vulnerability scanner will be used to improve the quality of the code and limit this risk

Auditors getting Administrator privileges or Administrator access to audits	
Type	Level

Damage	Medium (5). It can result in information leakage or be part of a larger attack
Reproducibility	Low (1). The two sets of users are separated
Exploitability	Low (1). Being part of one of the two groups does not give any advantage to elevate privileges. Audits do not contain usernames or passwords
Affected users	High (8). Administrators and Auditors
Discoverability	Low (1)
DREAD	Low (3.2)
Mitigation	No action will be taken

System Requirements

Storage space

User and experiment data will require less than 1Kb per record, therefore a few megabytes will be sufficient to store them.

Each measurement is expected to require at least 22 bytes. With 1 million measures per experiment, each experiment will require about 21MB of space.

field	type	size
Measure type	integer	2 bytes
Timestamp	timestamp with nano precision	8 bytes
Experiment id	integer	4 bytes
Measure	double precision floating point	8 bytes

CPU and memory

CPU and memory requirements will be determined with load testing after the initial deployment. The minimum allocation will be set to values able to sustain the expected average daily traffic. The aximum allocation will be set to values able to sustain 200% of the maximum expected traffic. Autoscale will be configured to follow the demand and contain costs.

GDPR Consideration

The application design requires only a minimal amount of personal information. All users will be able to retrieve, update and delete their own personal information, in compliance with GDPR. Complete deletion will preserve Staff Identification for traceability (GDPR, 2016).

An administrator will be able to assist users with their GDPR request.

Document	Field	Description
User’s record	Staff identification	Unique id number from the HR system
User’s record	Name	Given name(s)
User’s record	Surname	Family name
User’s record	Email Address	Professional email address
Experiment	None	
Measurement	None	
Audit	User’s staff identification	Only the user’s staff identification will be stored in the audit

References

- Aimar, A., Corman, A. A., Andrade, P., Fernandez, J. D., Bear, B. G., Karavakis, E., ... & Magnoni, L. (2019). MONIT: monitoring the CERN data centres and the WLCG infrastructure. In EPJ Web of Conferences (Vol. 214, p. 08031). EDP Sciences. Available from https://www.epj-conferences.org/articles/epjconf/pdf/2019/19/epjconf_chep2018_08031.pdf [Accessed 2/April/2022]

- GDPR - *Consolidated text: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)* – Art 17 – right to be forgotten (2016). Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02016R0679-20160504&from=EN

- Gillespie M. & Givre C. (2021) *Understanding Log Analytics at Scale*. 2nd Ed. O'Reilly Media Inc.

- Ghimire, D. (2020). Comparative study on Python web frameworks: Flask and Django. Available from https://www.theseus.fi/bitstream/handle/10024/339796/Ghimire_Devndra.pdf?sequence=2 [Accessed 2/April/2022]

- Khasawneh, T. N., AL-Sahlee, M. H., & Safia, A. A. (2020, April). Sql, newsql, and nosql databases: A comparative survey. In *2020 11th International Conference on Information and Communication Systems (ICICS)* (pp. 013-021). IEEE. Available from https://www.researchgate.net/profile/Mahmoud-

Alsahlee/publication/340978543_SQL_NewSQL_and_NOSQL_Databases_A_Comparative_Survey/links/5ec46445a6fdcc90d685d608/SQL-NewSQL-and-NOSQL-Databases-A-Comparative-Survey.pdf [Accessed 2/April/2022]

- Kubernetes (N.D.) Horizontal Pod Autoscaling. Available from https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/ [Accessed 2/April/2022]
- Kubernetes (N.D.) Translate a Docker Compose File to Kubernetes Resources. Available from https://kubernetes.io/docs/tasks/configure-pod-container/translate-compose-kubernetes/ [Accessed 2/April/2022]
- OWASP (N.D.) Cross Site Request Forgery (CSRF). Available from https://owasp.org/www-community/attacks/csrf [Accessed 2/April/2022]
- OWASP (N.D.) Logging Cheatsheet. Available from https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html [Accessed 2/April/2022]
- OWASP (N.D.) OWASP Anti-Ransomware Guide. Available from https://owasp.org/www-project-anti-ransomware-guide/migrated_content [Accessed 2/April/2022]
- OWASP (N.D) REST Security Cheatsheet. Available from https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html [Accessed 2/April/2022]
- OWASP (N.D.) Secure Coding Practices. Available from https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/migrated_content [Accessed 2/April/2022]

- OWASP (N.D) Threat Modeling Process. Available from https://owasp.org/www-community/Threat_Modeling_Process#stride [Accessed 2/April/2022]

- Rabiee, A. (2018). Analyzing Parameter Sets For Apache Kafka and RabbitMQ On A Cloud Platform. Available from https://www.diva-portal.org/smash/get/diva2:1232563/FULLTEXT01.pdf [Accessed 2/April/2022]

- RabbitMQ (N.D.) Client Libraries and Developer Tools. Available from https://www.rabbitmq.com/devtools.html [Accessed 2/April/2022]

- Reagan, R. (2018). Message Queues. In: Web Applications on Azure. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2976-7_9

- Souza, R. D. A. (2020). Performance analysis between Apache Kafka and RabbitMQ. Available from http://dspace.sti.ufcg.edu.br:8080/jspui/bitstream/riufcg/20339/1/RONAN%20DE%20ARAU%CC%81JO%20SOUZA%20-%20TCC%20CIE%CC%82NCIA%20DA%20COMPUTAC%CC%A7A%CC%83O%202020.pdf [Accessed 2/April/2022]

- W3Techs (2022) Usage statistics of Nginx. Available from https://w3techs.com/technologies/details/ws-nginx [Accessed 2/April/2022]