MyMONIT

Collecting measurements to monitor CERN's experiments

Table of Contents

Table of Contents

Word count: 1212

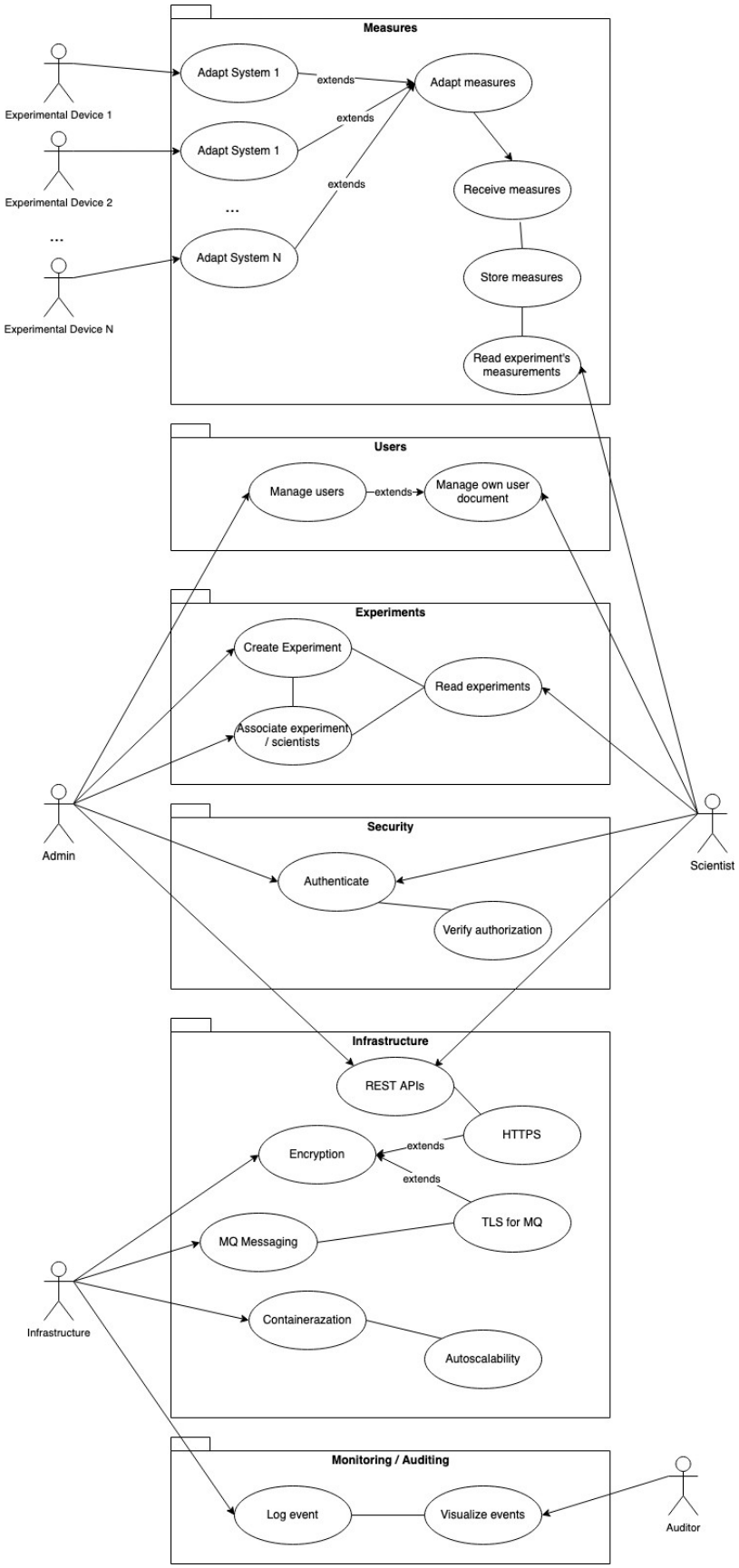
High-level description

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 monitoring to detect anomalies in the system itself and the flow of the measurements.

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	user's record	RW
	experiments	only records associated with user	RW
	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

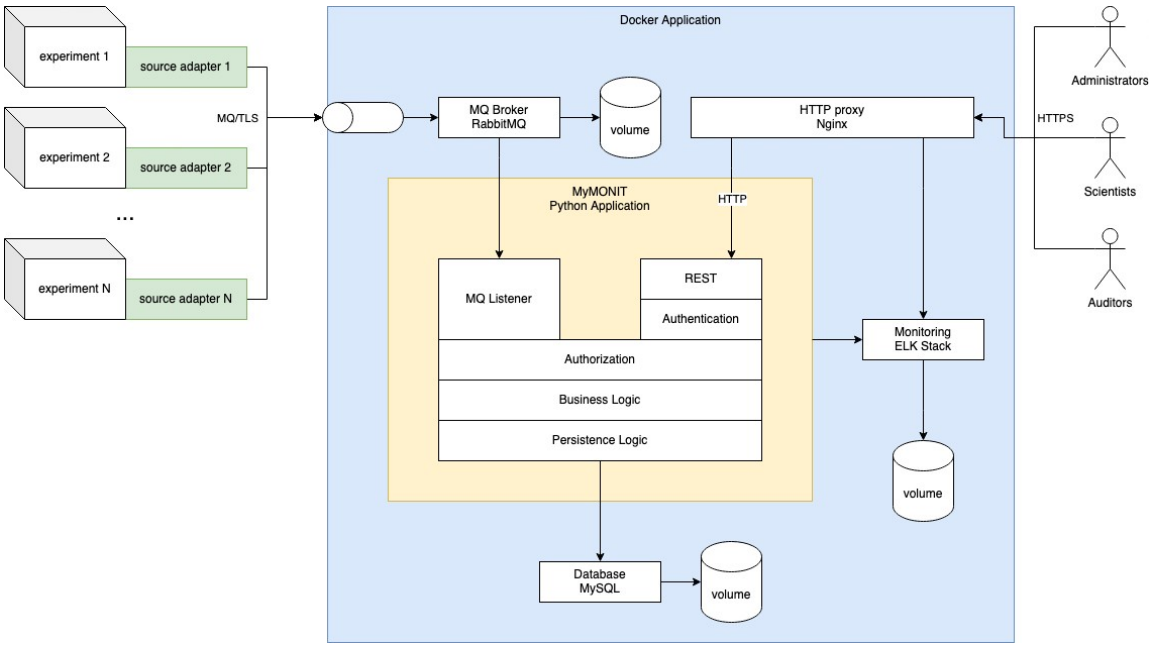
- Access points will be authenticated.
- Access to experiments will require per-user authorization.

- The system must serve concurrent users and concurrent experiments.
- 100% of data must be retained.

Assumptions

- The system's capacity must accommodate at least 10 years of data.
- Autoscale functionalities in the clustering infrastructure will be sufficient to regulate the number of running instances and deal with variable demand (Kubernetes, N.D. a).
- It is expected an elevated flow of data coming from queues. It is assumed that queues will absorb peaks of traffic (Reagan, 2018).
- Users will visualize measures polling the APIs. It is assumed that pagination will be sufficient to reduce the performance load.

Architecture

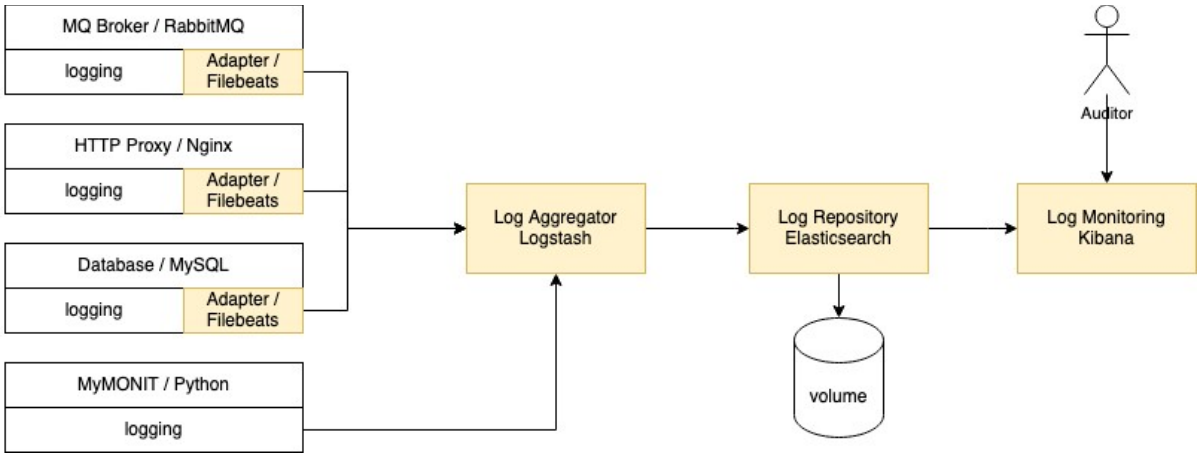


The the adapters (in green) will send the measurements to the solution (in blue) where the main component (in yellow) will index them and expose them via REST APIs.

- 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 to each specific case and will be installed at the experiment's location.
- Nginx will be used as 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 data streams from the experiments encrypted in TLS. RabbitMQ is a popular solution for this kind of applications and it war preferred to its competitor Kafka for its simplicity and

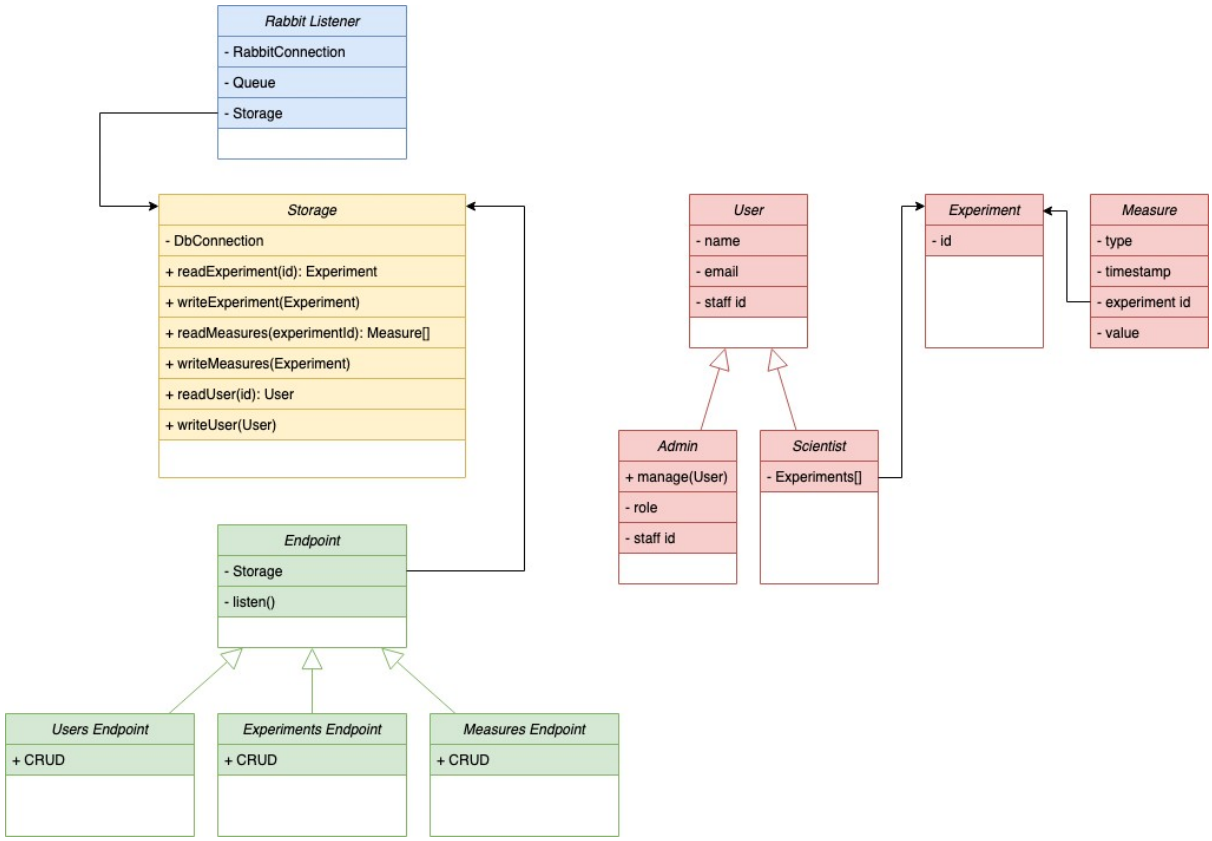
because it guarantees global message ordering when ran in a cluster (Souza, 2020) although Kafka offers a better scalability for high volumes of traffic (Rabiee, 2018; Souza, 2020)

- MySQL will be responsible for the storage of the application's data. A SQL database was preferred for its simplicity. The design will allow to replace it with a NoSQL database with minimal changes. NoSQL databases offer, in general, better scalability (Khasawneh, 2020).
- MyMONIT will be a Python application using Flask and Pika. Flask allows for rapid development (Ghimire, 2020). Pika is the first recommended library to support RabbitMQ in Python (RabbitMQ, N.D.)
- ELK Stack (Elastic Search, Logstash, and Kibana) will collect all logs and expose a dashboard for auditing. 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).



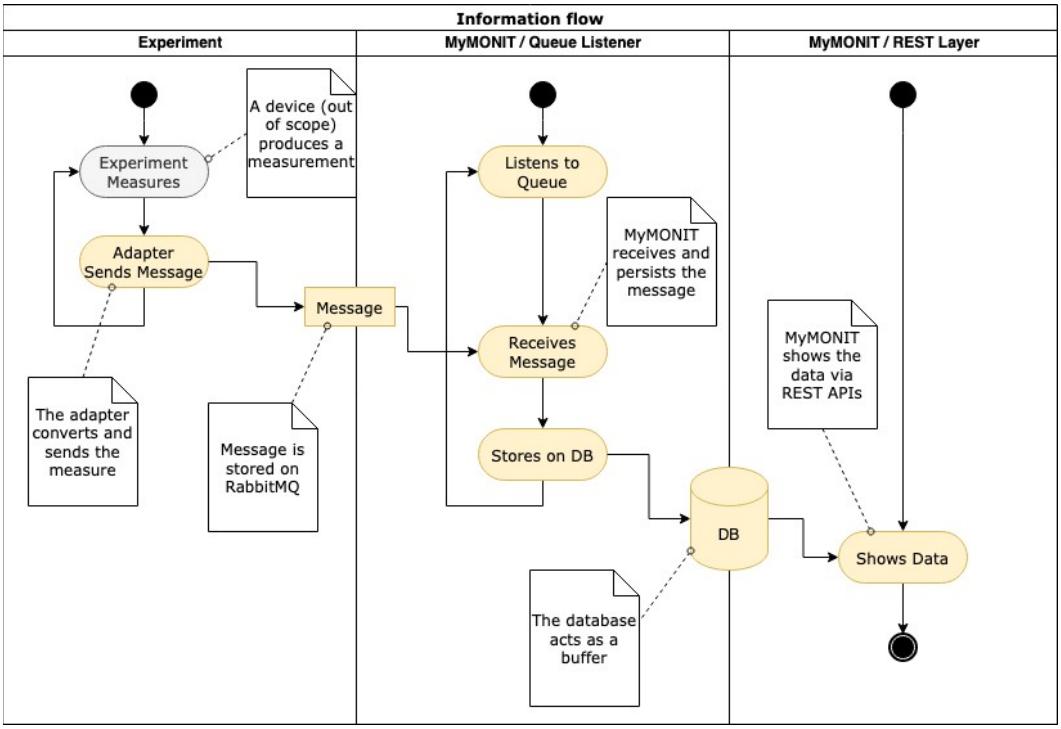
MyMONIT

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.

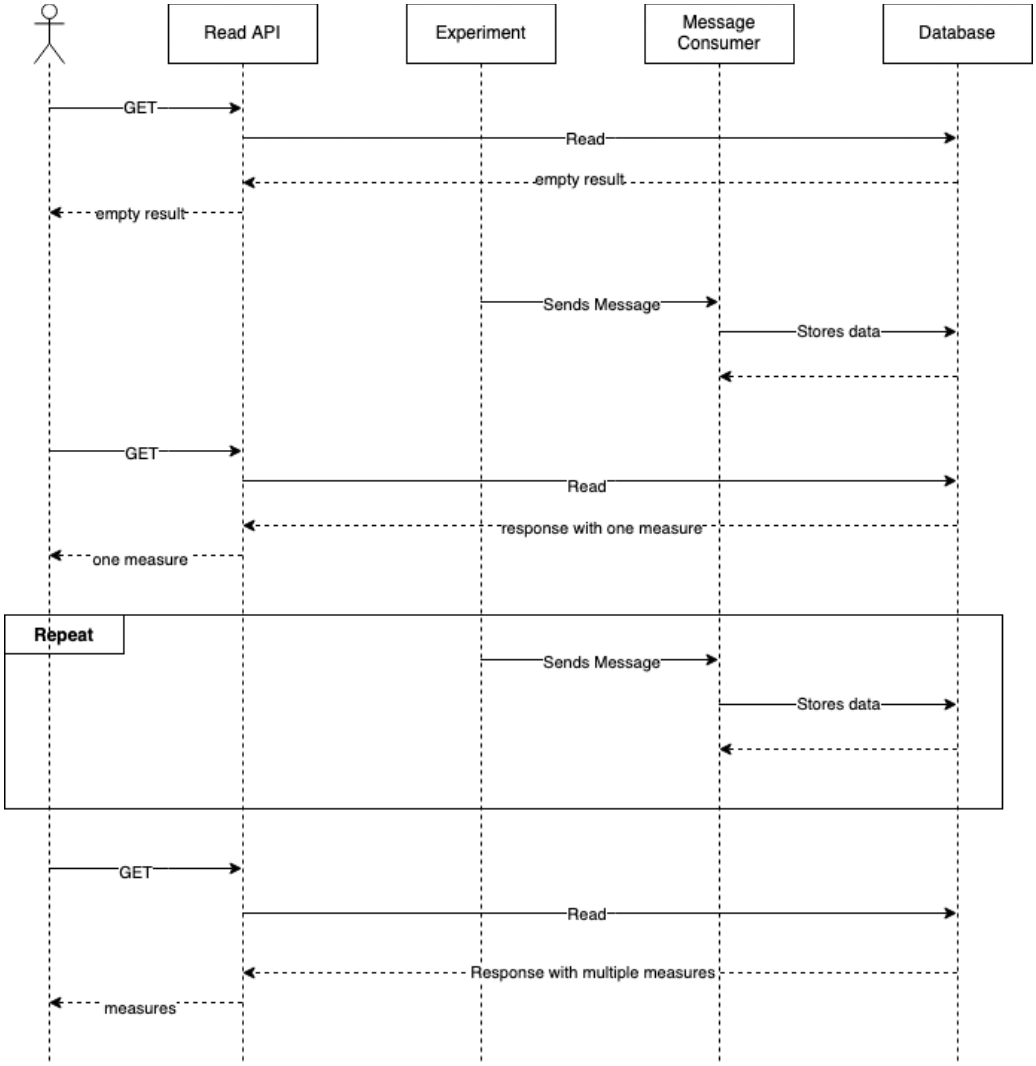


Information flow

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



The following represents the same flow from a timeline perspective


Read API
Experiment
Message Consumer
Database
GET
Read
empty result
empty result
Sends Message
Stores data
GET
Read
response with one measure
one measure
Repeat
Sends Message
Stores data
GET
Read
Response with multiple measures
measures

Security

Overview

The main security concerns are the risks of sabotage and information leak. Being a monitoring tool, an attacker may try to disrupt its operations to cover the main attack on an experimental facility. Information leaks could endanger the process of peer reviews allowing scientists to steal data from parallel research.

Authentication

The authentication endpoint will validate credentials by comparing the input with the hashes stored in the database. The endpoint will return a JSON web token that will remain valid for a limited time. The token will be required in the calls to all other APIs. The authentication endpoint will require the inclusion of a shared secret (API key) in the request. This additional measure will limit the chances to perform a brute force attack (OWASP, N.D.).

Authorization

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

Security Risks

Using the STRIDE model, the following threats were identified and classified with DREAD.

Spoofing

User's credentials violation

Type	Level
Damage	High, experiments would be exposed, users' records compromised, data leak
Reproducibility	High
Exploitability	High
Affected users	One user. All, if the user is administrator
Discoverability	Medium. User's credentials may be easy to guess

Tampering

Introducing fake measurements on the message broker

Type	Level
Damage	High, experiments would be invalidated
Reproducibility	Medium. The highest risk is broker's authentication
Exploitability	High. Discovering credentials would make it easy to exploit the vulnerability
Affected users	All scientists
Discoverability	Medium. The broker is public, but credentials are highly secure

Manipulate audits in Elasticsearch

Type	Level
Damage	Medium, it could be part of a more vast attack and it could delay the detection of an issue
Reproducibility	Low. It requires another violation
Exploitability	Low. It is hard to manipulate audits stored in Elasticsearch
Affected users	Auditors
Discoverability	Low. Elasticsearch is not directly exposed

Service disruption by deleting documents

Type	Level
Damage	Medium, data could be recovered through backups, activities could suffer delays
Reproducibility	High. Administrators could easily manipulate records
Exploitability	High. Administrators can manipulate records as part of their role
Affected users	All scientists
Discoverability	High. Administrators can manipulate records as part of their role

Repudiation

Administrator denies committing sabotage

Type	Level
Damage	Low. It affects only investigations after the fact
Reproducibility	Low. All actions are audited. Administrator do not have W access to audits
Exploitability	Low. Administrators do not have W access to audits
Affected users	All scientists
Discoverability	Low. It requires the discovery of additional vulnerabilities

Information disclosure

Database breach

Type	Level
Damage	High, data would be exposed
Reproducibility	Low. Database is not directly exposed, authentication is in place
Exploitability	Low. Attacker should compromise at least another system first
Affected users	All
Discoverability	Low

Scientists stealing measurements

Type	Level
Damage	Medium. Peer reviews may be invalid
Reproducibility	Low. It requires another violationLow. It requires another violation
Exploitability	Low. It requires another violation

Affected users	Scientists involved in the experiments, external stakeholders
Discoverability	Low

Auditors steal information through audits

Type	Level
Damage	Medium to High. Peer reviews may be invalid. Security may be compromised
Reproducibility	High. Auditors have access to audits as part of their role
Exploitability	High. Auditors have access to audits as part of their role
Affected users	Administrator, Scientists, and Stakeholders
Discoverability	High. Auditors have access to audits as part of their role

Denial of service

DDoS on APIs

Type	Level
Damage	High, system may become inoperative
Reproducibility	Low. The system should be exposed only in the internal network
Exploitability	Low. It would be easy to block the attack in the internal network
Affected users	All
Discoverability	Low. It would be difficult to plan an effective attack.

DDoS on Audit and Monitoring

Type	Level
Damage	Low to High. It may cover a more vast attack
Reproducibility	Low. The system should be exposed only in the internal network
Exploitability	Low. It would be easy to block the attack in the internal network
Affected users	Auditors
Discoverability	Low. It would be difficult to plan an effective attack.

Elevation of privilege

Scientists becoming administrators

Type	Level
Damage	High, the attacker could disrupt the system
Reproducibility	Low. It would require database access since no system function manipulates roles
Exploitability	Low. Attacker should compromise at least another system first
Affected users	All
Discoverability	Low

Auditor getting Administrator privileges or Administrator accessing to audits

Type	Level
Damage	Low to High. It can result in information leakage or be part of a larger attack
Reproducibility	Low. The two sets of users are separated
Exploitability	Low. 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	Administrators and Auditors
Discoverability	Low

System Requirements

Storage space

User and experiment data will require less than 1Kb per record, and their number is expected to be in the range of thousands. Therefore, it is safe to assume that 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. Minimum resources will be set to values able to sustain the expected average daily traffic. Maximum resources 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 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 30 March 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 30 March 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 30 March 2022]

- Kubernetes (N.D.) Horizontal Pod Autoscaling. Available from https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/ [Accessed 30 March 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 30 March 2022]
- OWASP (N.D) REST Security Cheatsheet. Available from https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html [Accessed 30 March 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 30 March 2022]
- RabbitMQ (N.D.) Client Libraries and Developer Tools. Available from https://www.rabbitmq.com/devtools.html [Accessed 30 March 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 30 March 2022]

- W3Techs (2022) Usage statistics of Nginx. Available from https://w3techs.com/technologies/details/ws-nginx [Accessed 30 March 2022]

Bibliography (TBD remove)

-

- Rabbit: https://www.rabbitmq.com/production-checklist.html