INTEGRATIVE DESIGN AND PROCESSING OF ENVIRONMENTAL IOT AND MICROSCOPY DATA VISUALIZATION

By

Ribhav Jain

Senior Thesis in Computer Engineering

University of Illinois Urbana-Champaign

Advisor: Professor Klara Nahrstedt

May 2021

Abstract

This thesis discusses the design and integration between the Clowder/4CeeD and Senselet frameworks. It speaks about the fusion and correlation of data in the MongoDB/Clowder system from the Senselet database InfluxDB in a secure and coordinated manner. Also, the thesis delves into research and study of Clowder/4CeeD and Senselet systems in addition to other similar data storage systems and techniques. These systems accelerate making scientific discoveries by allowing researchers to curate, store, and analyze their experimental data. The thesis will also delve into the technical details, architecture, and design of these systems in great detail while discussing and analyzing their implementation's various pros and cons. 4CeeD makes use of MongoDB for its heterogeneous data; meanwhile, Senselet uses InfluxDB to store time-series data. These cloud systems' advantages include fault-tolerance, availability, and efficient processing at scale, enabling them to revolutionize and improve the process of making scientific discoveries.

The time-series data stored in InfluxDB in the Senselet system contains environmental data obtained from external sensors in laboratories used by scientists for experiments. 4CeeD, on the other hand, stores experimental microscopy data in MongoDB, which scientists wish to correlate and compare with the previously mentioned environmental data. Together the two platforms form a sensory network architecture with a cloud backend that allows researchers to retain and correlate their data with ease in real-time.

This thesis goes into depth about the function and interaction between the two described systems, followed by the design discussion of a fusion framework developed to correlate the two. The framework allows scientists to easily correlate data between the two systems by visualizing the Senselet data stored in InfluxDB inside the Clowder platform. This enables researchers to study and analyze data from one platform within another. Also, researchers can extract and migrate data from InfluxDB (Senselet) to MongoDB (Clowder), allowing them to store environmental information with experimental data. Thus, this fusion framework allows researchers to visualize, correlate and migrate data seamlessly between platforms enabling them to improve and streamline their research process.

Subject Keywords: distributed systems; cloud; IoT; visualization; databases.

Acknowledgments

First and foremost, I would like to thank my research advisor Dr. Prof Klara Nahrstedt, for the excellent advice and support. Her experience and knowledge truly guided and helped me in all my work and research. I would also like to thank Todd Nicholson and Steve Konstanty for all their help and guidance in my research and development process. I would like to thank all Data Management Project Group members who constantly provided me with feedback and help. I have learned things from nearly every member's area of expertise and found everyone extremely helpful while conducting my research and development. Finally, I would like to acknowledge the research effort around 4CeeD funded by the National Science Foundation, NSF ACI 1835834 and Senselet by the National Science Foundation (award number 1827126).

Contents

1. Introduction

The environment consists of the 4CeeD and Senselet systems that lie under The Timely and Trustworthy Curating and Coordinating Data Framework (T2C2) that drastically reduce the materials-to-device process that can often span several decades. Semiconductor cleanrooms used in research to fabricate devices often deal with minute particles that must be dealt with extremely carefully and thus need experimental and environmental monitoring. The 4CeeD and Senselet systems allow researchers and scientists to collect, archive, analyze, and share collected digital data from labs and testing sites before archiving and publishing it for widespread usage [1]. These cloud systems exist separately and have very different methods of authentication, access control, and data storage. The Senselet system uses InfluxDB to store time-series data, while the 4CeeD (Clowder) system runs on MongoDB. The problem in the current state of materials-to-device and scientific experiment processes is that they are incredibly long, inefficient, and unorganized. In particular, the current state of data capture and storage in materials and semiconductor domains often involves many manual processes that lead to poor documentation of results. These systems accelerate the experimental process scientists follow by providing them with an efficient workflow tool to upload, store and manage microscopy data. They also leverage the advantages of shifting to the cloud, i.e., privacy, security, and scalability. The overall framework focuses on capturing, correlating, and coordinating the data obtained in real-time across various experiments and fields.

While these two data management systems have benefitted researchers and shortened scientific research and materials-to-device processes — they still lack smooth integration. Scientists and researchers have to manually correlate data between the two platforms in a highly tedious and inefficient way. They do not have a way to view the external sensor data for their experiments in the 4Ceed system and are therefore unable to detect anomalies or trends in an easy manner. Since being able to correlate the external and experimental data stored in the two different platforms is extremely helpful for researchers, this project integrates the two. The researchers currently use a sneakernet to transfer and correlate data manually, a tedious solution that is problematic and inefficient in their research domain. Since there is no integration between these platforms for researchers to analyze, compare and correlate data, this project is needed to improve workflows and efficiency. This thesis

delves into the approach developed to solve this problem and discusses the advantages and disadvantages of other potential approaches.

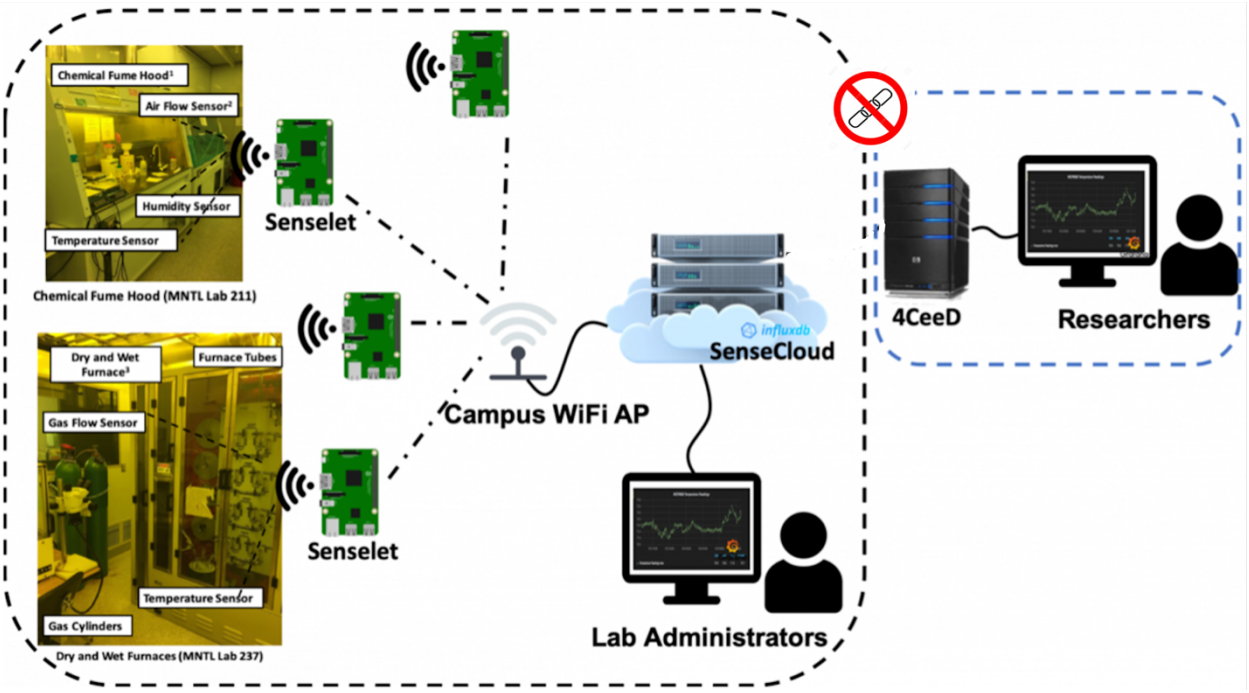

Figure 1 The separate Senselet (left) and 4CeeD (right) Systems (The goal of this project is to connect these two systems) [1]

This integration is accompanied by various challenges, such as different data formats and storage system types. 4CeeD uses MongoDB to store experimental data, whereas Senselet stores time-series data in InfluxDB. The two different data formats and databases pose a problem in the integration and require processing and migration. In addition, the data itself is highly varied as Senselet stores time-series humidity and temperature information from Semiconductor cleanrooms as opposed to 4CeeD's experimental microscopic data. In addition, the framework needs to function in real-time to provide scientists and researchers with up-to-date information about their experiments. Also, the integrative framework systems need to follow existing access control policies and function securely and reliably. These points present the challenges that need to be tackled in the development of an end-to-end integrative framework that solves the problem at hand.

Related work includes modern-day cloud systems such as DropBox and Google Drive that speed up data capture and storage by eliminating the manual processes and poor documentation. Traditionally, researchers would handwrite data in books or manually transfer data with the help of flash drives. These methods were not only slow but also inefficient. They also introduce constraints such as how scientists could save data locally, on their flash drive, or in writing. This also led to implicit bias wherein researchers only kept the 'best' results that they felt were most important to their experiments. Systems such as NanoHub and DataUp focus more on data availability; however, 4CeeD and Senselet offer an end-to-end process solution specifically built to aid researchers and scientists. Having a cloud system that captures and curates all data in real-time eliminates any recording-related constraint by ensuring the storage of all relevant data. As mentioned previously, these systems also introduce fault tolerance and prevent data loss with the failure of a personal laptop or hard drive. Also, cloud systems provide high availability that allows researchers to access their data stored in the cloud from any device anywhere in the world. Thus, the 4CeeD and Senselet cloud systems offer a convenient and efficient solution to researchers' problems safely and securely.

While currently there does not exist an ideal solution to integrate two platforms, there are some approaches that can be adopted. Data can be manually transferred from both platforms into a common directory using cloud systems such as DropBox and Google Drive. This periodic manual approach does not promise up-to-date data and is tedious for researchers. The data from both platforms are in different formats and, therefore, challenging for users to make sense of in the raw format. In addition, it adds the hassle of using a third platform and poses access control risks. Another approach is the use of a memory stick to store and correlate data from both platforms. This also poses the issue of not having up-to-date data in addition to being a security threat that can be lost/stolen. This thesis describes a suitable approach that gives a more holistic view of the experiment since it ensures the retention of all data with smooth integration service to analyze and correlate data. It also enables swift and smooth sharing and transfer of data [7][9].

Ultimately, the project involves using an iframe to visualize data from Senselet with the help of Grafana inside 4CeeD. The approach solves the problem in question in an efficient and user-friendly manner while also being easily maintainable. The visualization of environmental data helps scientists better understand and diagnose their experiments within the comfort of Clowder/4CeeD. Researchers can securely view their data, spot anomalies, correlate datasets, and visualize information in a coordinated manner. They no longer need to constantly switch between platforms and can view data from one

platform inside another. In addition to visualizing data, researchers can also download the data in a JSON format or as a CSV file and store it with relevant datasets. Another feature offered to researchers gives them the ability to extract and migrate data seamlessly from one platform to another. Researchers can provide a data range within 4CeeD that an extractor uses to query Senselet's InfluxDB and migrate data. This enables users to access the environmental data they want within the 4CeeD platform in the relevant dataset. This feature set further discussed in this thesis enables researchers to easily view, analyze and migrate data between platforms simplifying and expediating the scientific research process.

The results in creating this fusion framework were positive as the development was completed and demo-ed to relevant researchers in a timely manner. There was positive feedback from the scientists who found the framework helpful in their research processes and gave helpful feedback to improve the system. The project spanned over approximately nine months and consisted of multiple phases. After researching and understanding the 4Ceed/Clowder and Senselet systems, I spoke to researchers and scientists to understand their problems. Once I realized the issue at hand, I started designing an integrative framework that could solve it. The main challenge involved was to strike a balance between intrusive software development and ease of use. After some research and prototyping, a design was finalized that met the scientists' needs and was also simpler to develop while being low maintenance. The design kept in mind the protocols, programming languages, and frameworks that 4CeeD uses to ensure a smooth integration. I then worked to develop the framework according to the design and test each feature. Once the development was done, I tested and demoed it to the relevant stakeholders. Finally, I evaluated the platform and documented the design and metrics.

2. Background

2.1. 4CeeD

The 4CeeD system leverages the Clowder framework developed at the National Center for Supercomputing Applications and consists of two data blocks called The Curator and Coordinator. Clowder is an intelligent data management system that helps capture, curate, and correlate scientific data in real-time [1]. It can support any data format and is a scalable and customizable data management framework. The flexibility of Clowder allows both user and machine-defined metadata, including the ability to enter directly from the User Interface [10]. The framework focuses on capturing, correlating, and coordinating the data obtained in real-time across various experiments and fields. The entire scientific process starting from the curation of data to correlation and coordination takes a long time and is too tedious for researchers. To circumvent this tedious and cumbersome data collection process and sharing, scientists can use the 4CeeD system and conveniently share their discoveries. Besides, a cloud storage service such as 4CeeD also enables future scientists to build upon the research work done earlier in a simple manner. The cloud infrastructure for 4CeeD is similar to Dropbox or Google Drive, wherein users can easily manage, store, analyze, and annotate their data. Through efficient data flows, scientists can use 4CeeD to get real-time results and measurements from their experiments and act accordingly. This cloud system also enables scientists to upload images, tag files with metadata, and receive insights about their findings. The simple yet efficient access to all information provided by 4CeeD makes it an excellent application for researchers worldwide while providing the benefits of privacy, security, availability, and scalability [2][9].

4CeeD also leverages the advantages of shifting to the cloud, i.e., privacy, security, and scalability. The system has a secure access control regime that ensures data is only accessed by those with the permissions to do so. The system also boasts availability that allows researchers to access their data from any of their devices anywhere in the world. This feature provided by modern-day cloud systems allows scientists the flexibility and convenience of not having to worry about the failure, theft, or absence of a personal device since their data is already backed up. The system is also moving towards a mirror storage system that can potentially deal with server failures by backing up data, thereby providing high availability and fault tolerance [2][9]. As mentioned previously, 4CeeD allows the secure

upload, organization, and description of data through a simple 3 step process: 1) Organize their data in collections and datasets while attaching metadata to it. 2) Quickly create datasets from custom-created templates and also share them. 3) Securely upload various files of multiple types easily. 4CeeD also adds new features such as updated dashboards, LDAP Authentication, and Jupyter Notebook Integration. These features make it much easier for users to be able to visualize their data and gain insights. LDAP Authentication is another feature that helps 4CeeD increase its security and access control methods and make integration easy for organizations [2]. Breaking the 4CeeD system down; it is broken into two data blocks: the Curator and Coordinator. The Curator service is responsible for acquiring data in real-time from the various sensors and measuring instruments from the labs. Currently, at the University of Illinois, the Material Research Lab and Micro-and-Nanotechnology Lab use this system to curate the digital data obtained in the process of collaborative research. This data can also be linked with meta-data and other relevant information in real-time in a trusted manner without direct researcher intervention. The Coordinator data block is where the data is filtered and made sense of. Various data processing techniques are used to find correlations and dependencies in the data that help researchers better understand their work. Ultimately, these two blocks significantly reduce the effort and development time, and cost involved in the end-to-end research process [1][2][7].

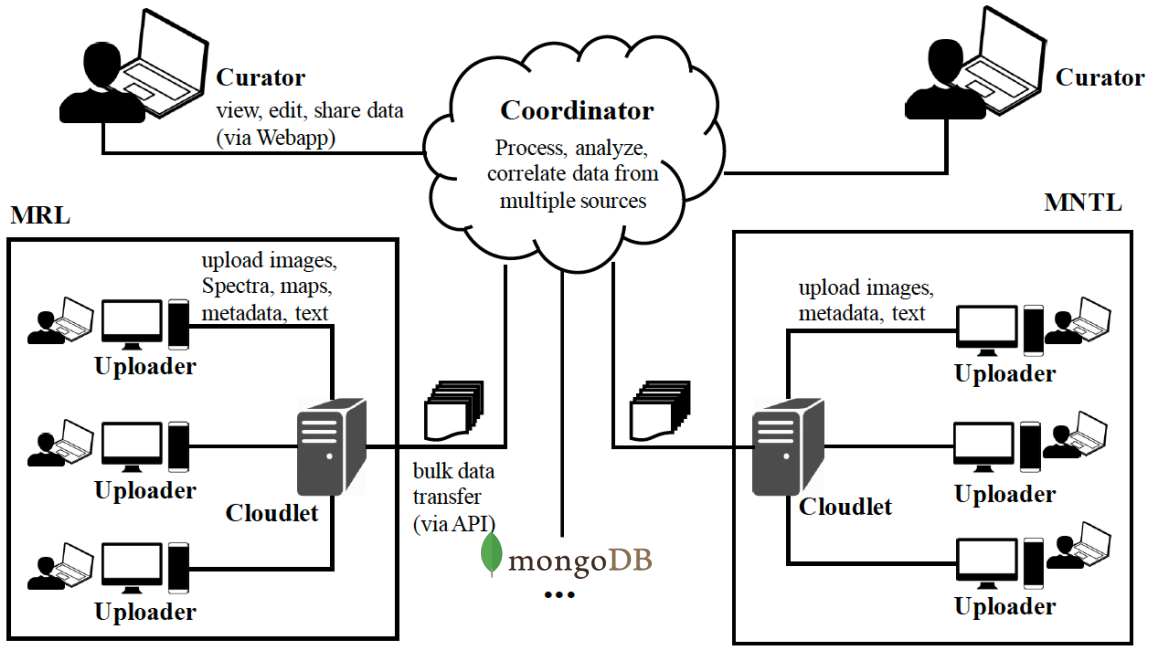

Figure 2 An overview of 4CeeD [7]

The technical design and architecture of the multi-tier 4CeeD infrastructure consist of a curation, cloudlet, and coordination service. The curation service performs adaptive data collection from the researcher's experimental instruments and wraps them with metadata in real-time. Users can upload the raw data generated from instruments and add any relevant notes to help describe the process. This process of curation is not only thorough with extensive information but also secure and trusted. In the next tier, the cloudlet service caches the data obtained from the curation service and facilitates data transfer with the cloud service's back end. This tier's caching aspect helps avoid traffic congestion and overload while scheduling data transfer and checks for emergency alerts after pre-processing the data. This intermediate tier is optional if the scientific instruments are connected to secure PCs that are patched to remain updated and secure. The coordination service is the centralized part of the cloud infrastructure for 4CeeD.

The storage and processing of data happen in this stage and are based on the cloud-based pub-sub system. This tier finally filters, extracts, processes, and analyses the data coming from multiple different sources. It also finds correlations in the data and other dependency relations that may occur between materials in the fabrication process. The data model used by 4CeeD includes the three main concepts of nested collections, datasets, and files. Users can organize their datasets in multiple collections and form a nested structure that is easy to navigate and understand. 4Ceed has an intuitive user interface, including a web-based app for the uploader and simple screens to view, edit and search experimental data [1][7].

2.2. Senselet

Senselet, on the other hand, is a Sensory Network Infrastructure for Scientific Lab Environments. It is an intelligent distributed sensing system that's used for scientific cleanrooms. The environmental conditions in a research project are critical, and Senselet is a cloud system that monitors and tracks this data. Semiconductor cleanrooms used to fabricate devices often deal with sizes smaller than dust particles and must be dealt with extremely carefully. As next-gen materials like transistors and LEDs become smaller, it becomes even more crucial to track and monitor their production environmental conditions. Often such cleanroom environments are un-monitored and un-controlled and can lead to unfavorable experimental conditions. Ultimately with the help of Senselet, researchers can correlate

their experimental results with the environmental records. Through the monitoring offered by Senselet, labs can have safer research environments with fewer mishaps, such as instrument overheating and gas leaks. A central cloud system like Senselet allows for a great deal of preventative maintenance wherein scientists can notice trends that warn about possible future mishaps. Future paragraphs will discuss the use of IoT and Machine Learning in Senselet to predict environmental changes. Another advantage of Senselet is that it is an affordable solution for academic labs that do not have the budget to afford commercial grade equipment and solutions [1][3][4]. Senselet leverages scalable and reliable sensory infrastructure in academic cleanrooms in a scientific way by using IoT. With machine learning, Senselet can help analyze the environmental data and diagnose any issues in the experiments being conducted. It provides reliable services to monitor and control lab environments for lab managers while being affordable.

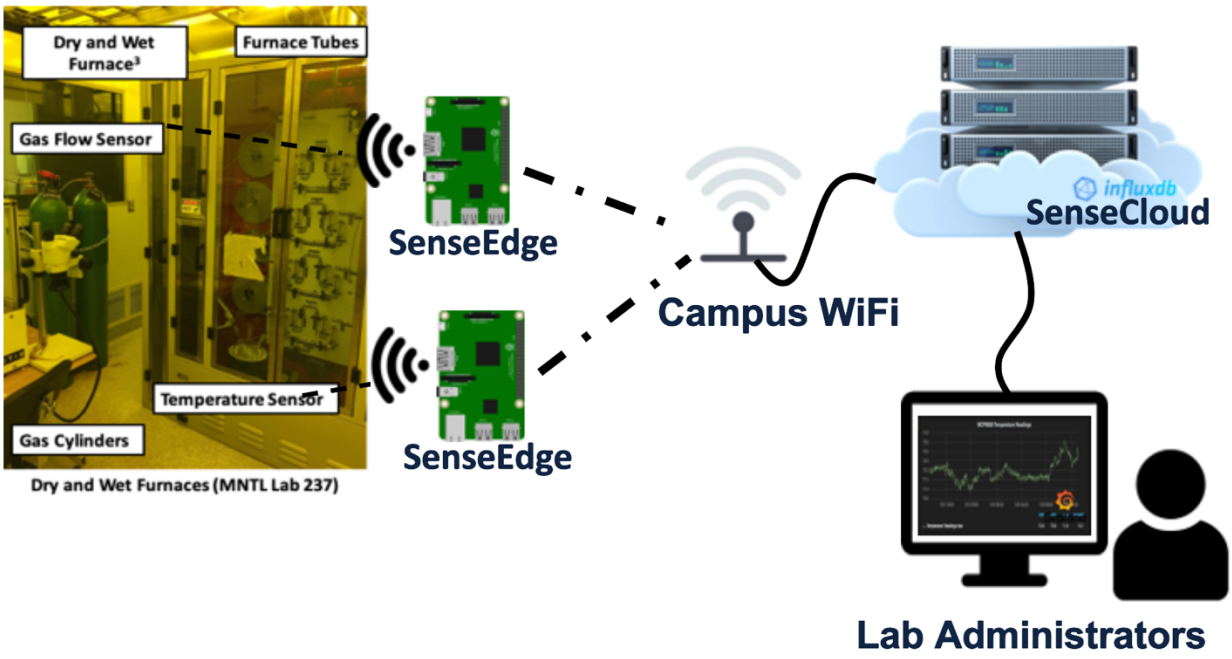

Figure 3 An overview of the Senselet system [1]

The sensory data around the experimental instruments is critical as information such as humidity, temperature, and vibration can be crucial for research in nanofabrication and biomedicine. A system that offers convenient upload, storage, and processing of data can significantly benefit researchers.

Another great feature offered is that of real-time emergency alert that can inform researchers of any situation that may lead to instrument damage. Examples of experimental data are microscope images taken in the laboratory, while metadata is the microscope settings corresponding to those images. Having all such data available makes it much easier to analyze and conclude results by correlating the various parameters during the fabrication process. Another reason the preventative maintenance and monitoring offered by these systems is essential is the sky-high cost of specific scientific instruments such as the Atomic Force and Scanning Electron Microscopes. Data can point to trends that show wearing down parts and thereby effectively avoid costly failures and mishaps. These sensitive devices are incredibly susceptible to changes in their environment and need constant monitoring [1][4][8].

The Senselet project aims to deploy wireless and scalable sensory infrastructure in experimental labs to measure and monitor external data for specific instruments. Ultimately Senselet allows us to seamlessly correlate and synchronize the sensory data with the instrument data (and metadata) in real-time, allowing better monitoring and control. This project aims to speed up the development of new innovations while freeing scientists from time-consuming, redundant experiments in the process. The use of IoT sensory networks paired with cloud infrastructure is a fantastic example of how we can leverage technology to foster innovation. The information provided by Senselet on the present status, trends, and predictive paths makes the discovery of new processes or bugs much more convenient. The system can also help detect anomalies and thereby promptly react and alert labs in case of emergencies [3][8].

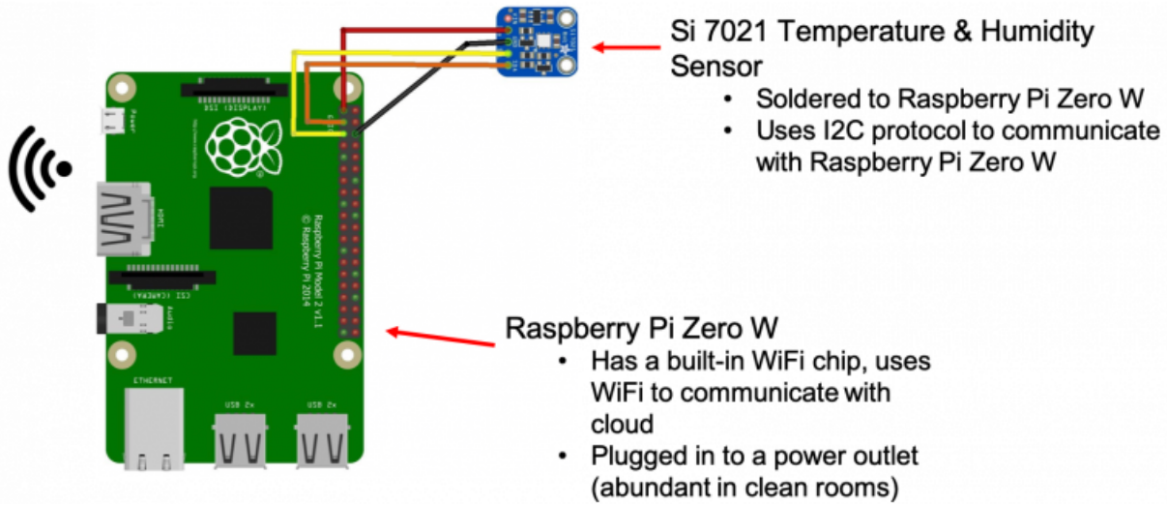

Figure 4 An overview of the Senselet Edge sevices with sensors [1]

The technical design and architecture of Senselet consist of a two-tier edge cloud architecture stored in the cloud and monitored by lab managers. The Senselet infrastructure includes various wireless sensors that can measure humidity, temperature, and vibration. These state-of-the-art sensors form an Internet of Things Sensory network that curates and stores their mission-critical data in the Senselet cloud system in real-time. The infrastructure also includes wireless edge devices that support multiple protocols such as Near Field Communication (NFC), Bluetooth Low Energy (BLE), and WIFI. These devices are placed close to the sensors and allow them to communicate their data to the cloud service with wireless communication protocols. These devices called SenseEdge consist of a Wi-Fi-equipped Raspberry Pi and commercial sensor soldered together. Once these devices obtain the sensors' data, they pre-process and upload them to the cloud service. The pre-processing phase includes any immediate emergency alerts such as intrusion detection or impending failures. Lastly, the system includes the cloud service called SenseCloud, which collaborates with 4CeeD to store and correlate the sensory data obtained in real time. On the cloud side, the database used is InfluxDB that stores time-series data.

A comparison between the databases used for 4CeeD and Senselet will help understand why this particular choice was made in future paragraphs. The visualization tool Grafana is provided to users to monitor better and judge the sensor data stored in a time-series manner in InfluxDB. The sensors' distributed system uses a heartbeat algorithm to detect sensor failures and uses the watchdog mechanism to achieve the self-reset feature. This monitoring service is highly available as a result, as needed by the cleanroom administrators [1][8].

3. Design

The two separate systems (4CeeD and Senselet) need a framework for integration that allows researchers to correlate the data between the two conveniently. Currently, scientists need to open the two systems independently and analyze data separately. Integrating the two systems will allow scientists to analyze and correlate experimental data with external data in a simple way. The objective to correlate time-series external sensor data with the internal microscopy data can be achieved in a few different ways.

3.1. Design Approaches

1) Migrate data

The first approach is to migrate data from Senselet stored in InfluxDB to the MongoDB instance from 4CeeD. This migration will allow the 4CeeD system to store the external sensor data in addition to the internal microscopy data. The 4CeeD system then contains all relevant data and can allow users to visualize their data. This approach ensures that only those with permissions to access 4CeeD will view and visualize the data without violating any current access control protocols. Since the data is transferred directly to a dataset, it follows the access control permissions for the same and is therefore secure. This also reduces the learning curve for scientists as they have access to both data segments in a single system since 4CeeD is highly flexible with the data it can store. The main drawback with this approach is the intrusive nature of the migration, wherein large amounts of data may have to be moved from one database to another. However, if researchers can select a date range for the environmental data they would like transported, then the inefficiency of excess data migration is minimized. However, if this approach is taken up alone, then the process of visualizing this data in the 4CeeD system will require a lot more work in addition to that of the data migration. This approach can still be developed efficiently with the use of extractors in 4CeeD, thereby minimizing the drawbacks, an approach we will discuss later in this section of the thesis.

2) Use an iframe to integrate the two systems

An iframe is an HTML tag that specifies an inline frame. It allows us to embed another document within the original HTML page called the parent browsing context. The iframe HTML tag also allows developers to specify global attributes such as height, width, margins, scrolling ability, and feature policies such as access to the microphone and camera [11][12]. Thus, an iframe allows us to display another webpage within Clowder, thereby allowing users to view and interact with a different platform and context. This approach uses this iframe component inside 4CeeD, thereby allowing researchers to view and interface with data from Senselet. Researchers can use this iframe to view, analyze, and correlate data between the two platforms without manually navigating different platforms. The iframe displays Grafana dashboards that visualize data stored in Senselet's InfluxDB and allow scientists to analyze trends and anomalies.

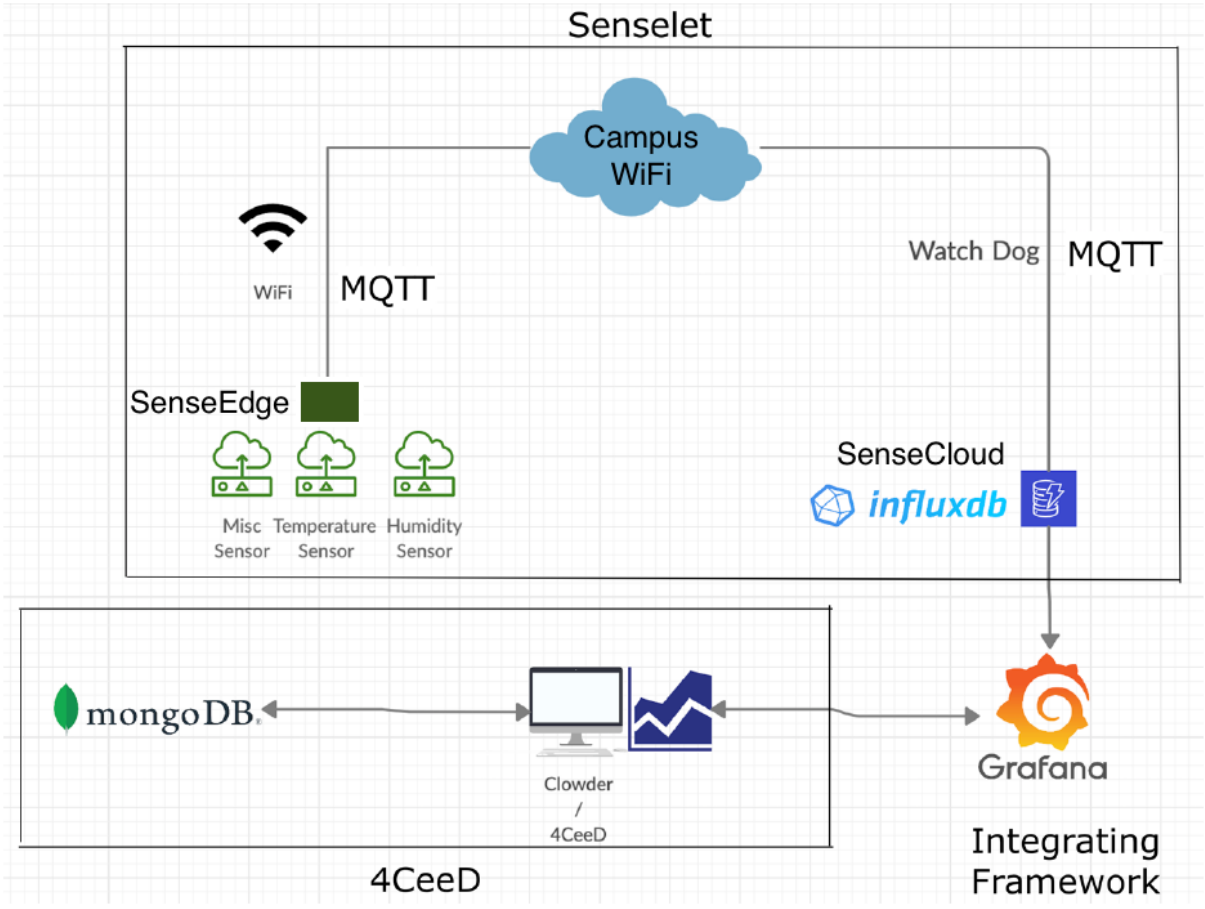

Figure 5 Data fusion framework design with an iframe

Grafana is an incredibly popular multi-platform technology used to display open-source analytics and interactive web application visualizations. It can create comprehensive monitoring dashboards that visualize data as specified by the specified interactive query. Grafana works extremely well with time-series databases such as InfluxDB that Senselet uses [13]. Thus, embedding a Grafana dashboard from Senselet's InfluxDB within 4CeeD is an extremely practical approach for our goals. This approach circumvents the need for any data migration and is, therefore, less intrusive. It requires less overhead work since the visualization is already done by Grafana and does not need to be done from the ground up. This approach also allows researchers to directly download time-series data in the raw format from the iframe and save it for their use. Since this iframe is embedded in a page inside 4CeeD, this ensures that there are no violations of access control. The data security is upheld as only those users with permission to access the data in 4CeeD can view this visualization of Senselet data.

3) Use an intermediary Cloud Storage System

Cloud Storage systems such as Google Drive and DropBox can be used by researchers as an intermediary platform. They can copy data from Senselet and 4CeeD into Google Drive or DropBox and analyze and correlate over there. While systems like DropBox and Google Drive may seem like an easy solution to use in between two separate storage systems, they have a few drawbacks. The addition of another platform increases the complexity and tediousness for researchers. In addition, the process of manually transferring data to this third platform is time-consuming and tiresome for scientists, thereby prolonging their research process. The data formats from the platforms are not similar either and will thus take a lot of effort from researchers to analyze in their raw format. Since this approach requires manual migration, the data in Google Drive and DropBox will not be up to date, unlike the Senselet system with real-time data.

4) Leave the platforms as they are

This approach keeps things as they are and requires researchers to navigate two different systems manually. They will need to use 4CeeD for experimental data and Senselet for environmental data before manually correlating the data from the two. This is an inefficient approach and requires much overhead work from researchers, which is far from ideal.

3.2. Adopted Design

The design approach finally selected was a combination of the iframe and data migration approach. 4CeeD/Clowder offers a feature wherein extractors, and external clients can attach metadata to files and datasets using the Web service API [10]. A user has the ability to trigger an extraction service in 4CeeD/Clowder that can process the data and add metadata to the dataset. We use this feature to enable users to use the extractor developed by us by providing a start and end date parameter in the Clowder interface. The given extractor then uses the given start and end dates to query the Senselet data stored in InfluxDB. Once the extractor has the query response from InfluxDB containing the temperature and humidity data for the given date range, it uses PyClowder to upload it to 4CeeD/Clowder. PyClowder is a Python-based library that makes it easy to create extractors and interact with Clowder in Python. The library is based on the Python Requests library that is standard for making HTTP requests in Python. Both PyClowder and Python Requests make it easy to write code that interacts with services and consuming data without worrying about the complexities of the communication [14]. The code in these extractors also contains functions that wrap the Clowder API and map to the routes endpoint of Clowder. Since the user performs the extraction of a certain dataset within Clowder that they have access to, the service uploads the query response as metadata to the very same dataset. This is a smooth process as PyClowder uses connectors such as RabbitMQConnector to upload the metadata back to the dataset from which it was requested [15]. Specifically, to implement this extractor, we have used a Python 3.7 script that makes use of PyClowder to query and process data from Senselet's InfluxDB for a given date period before uploading the data to 4CeeD/Clowder. This script pre-processes the temperature and humidity sensor data it gets from InfluxDB and converts it into a JSON meta-data format before uploading it to Clowder.

Another advantage of this approach is that it does not require any modification to the front end. It functions as a backend extractor service that Clowder supports and has already integrated into its interface. Thus, this approach helps integrate the data between the two platforms, giving researchers access to the environmental sensor data in the 4CeeD/Clowder platform that helps improve their research workflow. To ensure a smooth holistic integration that scientists can use for a broad range of use-cases, we have also implemented the iframe approach. As mentioned previously, an iframe is an HTML that specifies an inline frame that can embed another document within the original page. The iframe displaying Grafana dashboards that visualize Senselet's environmental data is an excellent way

for researchers to analyze and correlate data visually. Grafana also provides an easy way to filter and interact with the data through built-in features. Users can toggle the dates for which they wish to see data and use this cursor to hover over different data points and lines to see more.

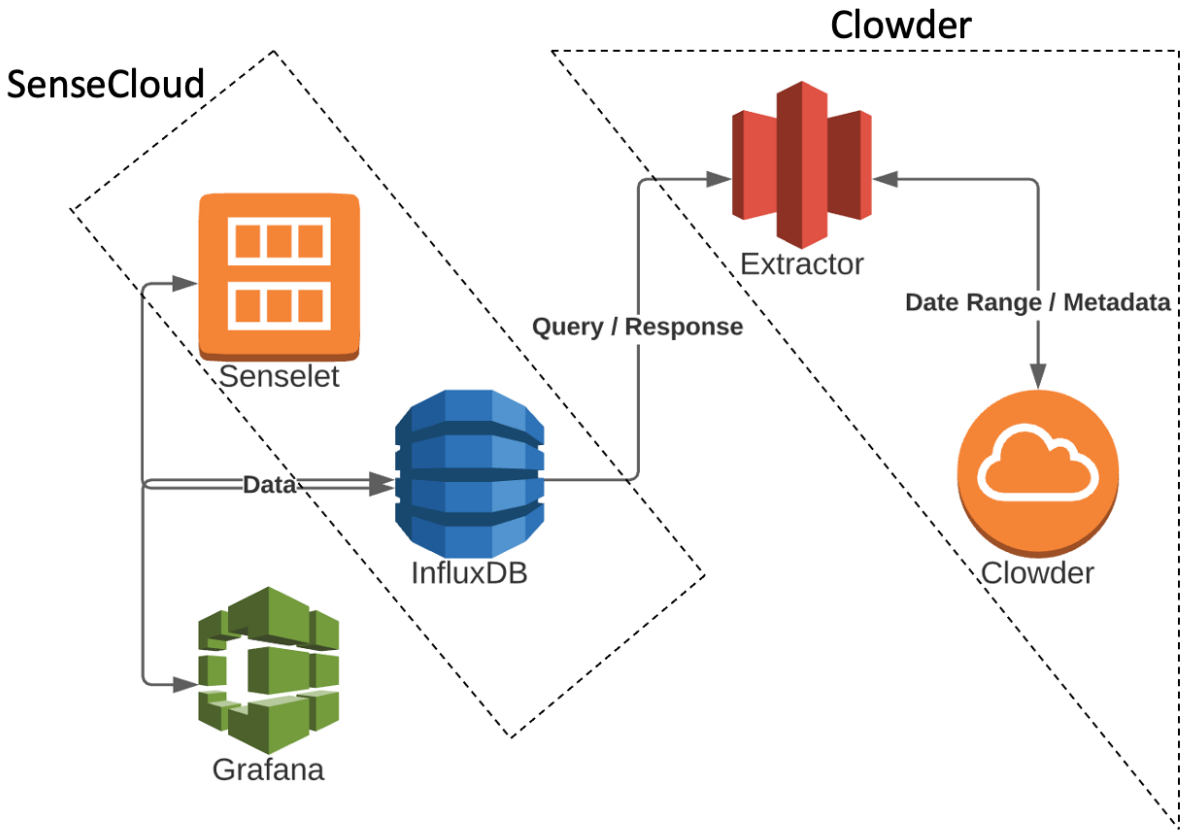

Figure 6 Data migration using an extractor

The Grafana dashboard also allows users to get shareable links to the dashboard they view and take screenshots for their records. Also, they can use specific options in the interface to download the data in a JSON format. Thus, using an iframe to embed a Grafana dashboard displaying time-series data from Senselet within Clowder is an efficient approach to help scientists. The extractor and iframe together form a holistic solution that allows users to visualize and analyze data using Grafana and then transfer relevant metadata into their datasets using the extractor.

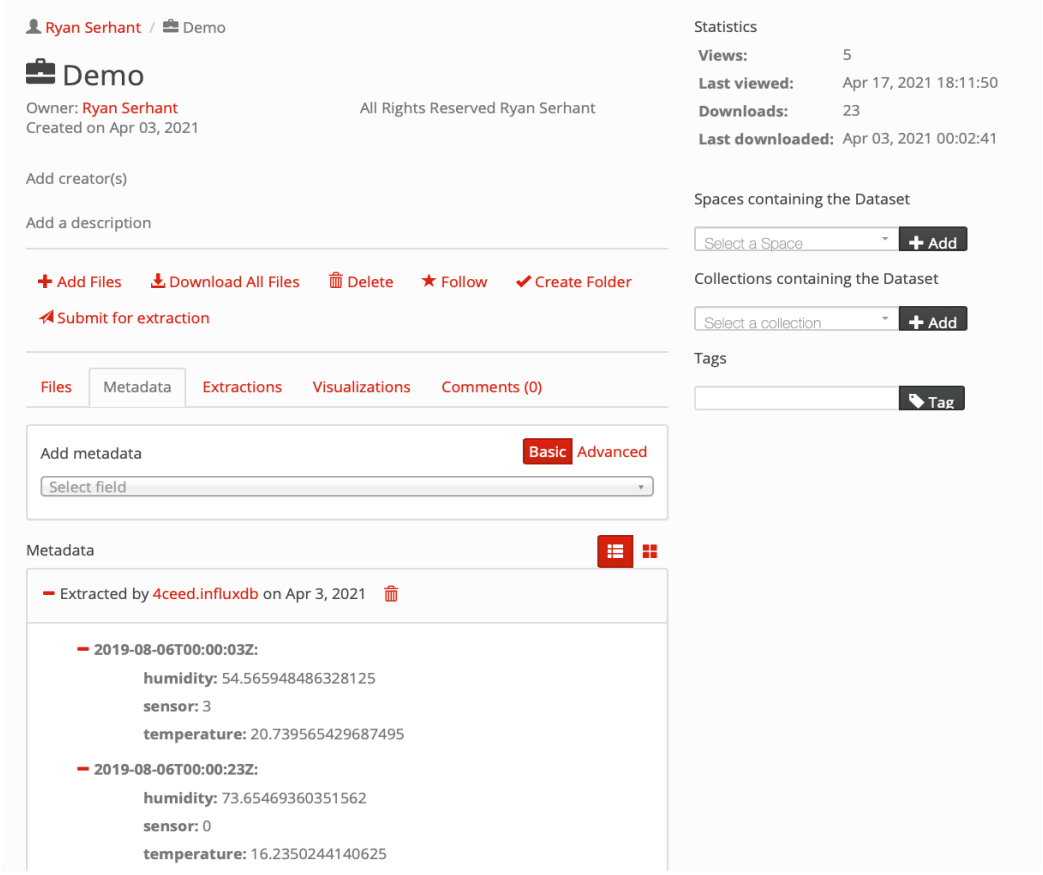

Figure 7 Environmental metadata uploaded to the Clowder dataset

4. Implementation

The implementation can be found on GitHub (https://github.com/ribhavjain/Integrative-Design-and-Processing-of-Environmental-IoT-Microscopy-Data-Visualization); in this section we will describe high-level functioning of the code. The extractor aspect of the fusion framework has been implemented using a Python (version 3.7) script. An extractor allows us to retrieve, process, and migrate data between platforms in the cloud. It utilizes libraries such as logging, JSON, and DateTime to perform the pre-processing and other standard extraction features. The script also uses the PyClowder (version 2.4.0) library described in the previous section. PyClowder is a Python-based library that simplifies the creation of extractors in Python that interact with Clowder. The library is based on the Python Requests library that is typically used to make HTTP requests in Python. Both PyClowder and Python Requests make it easy to write code that interacts with services and consuming data without worrying about the complexities of the communication [14]. In addition, the extractor script also uses the influxdb (version 1.8.3) python library that enables it to query Senselet's InfluxDB for environmental data within the provided dates. This extractor interacts with Clowder using the PyClowder library (based on Python requests and HTTP) and InfluxDB using the influx-python library (also based on Python requests and HTTP) [16]. The code in such PyClowder extractors also contains functions that wrap the Clowder API and therefore map to the routes endpoint of Clowder that is written in the Play Framework. The extractor has a function called process_message that takes a few parameters, including the connector, host, secret key, resource and parameters given by the caller. The parameters include the start and end date for the data to be fetched, while the secret key and resource aid in identifying and uploading to the dataset. Using the publish-subscribe or pub-sub model and the RabbitMQ (version 3.8.8) broker, the extractor can listen to calls and use PyClowder to write back the relevant information once it is done running [10]. When called, the extractor processes the message and calls the fetchInfluxDBPayload function with the required parameters. These parameters include the start and end time in addition to the username, user key, and password for the InfluxDB instance. The function then uses the parameters to create an InfluxDB query that will return the data for the given date range. Once influx.query is called, and the result is obtained, the script processes it and converts it to a Python dictionary data structure. This dictionary data structure uses time as the key, with the value being the humidity and temperature data. This data structure is then returned as a payload to Clowder as the metadata for the given dataset that is stored in MongoDB (version 3.6.19).

The iframe aspect of the framework has a more straightforward implementation as it is embedded within the HTML code for Clowder. Clowder uses the Play Framework, a full-stack framework with all the components required to build a Web Application. Play integrates the components and APIs we need in a stateless and efficient web-friendly architecture. The views in the Play framework are what users see on the front end and are written in HTML, i.e., Hypertext Markup Language—adding an iframe in a new page required the creation of a new view in the Clowder format with the standard headers and footers. The inline iframe tag was then added with a link to the Grafana dashboard that visualized Senselet data. As stated previously, an iframe can also be accompanied by several attributes such as height and width that specify the appearance and functioning of the frame. These features were implemented and tested on multiple browsers such as Safari, Google Chrome, and Mozilla Firefox. Once these features were implemented, they were tested to ensure stability and functionality before being demonstrated to the relevant stakeholders. Adding the inline iframe tag embedded the Grafana dashboard seamlessly into the Clowder interface and allows users to visualize, analyze and download Senselet data from this window.

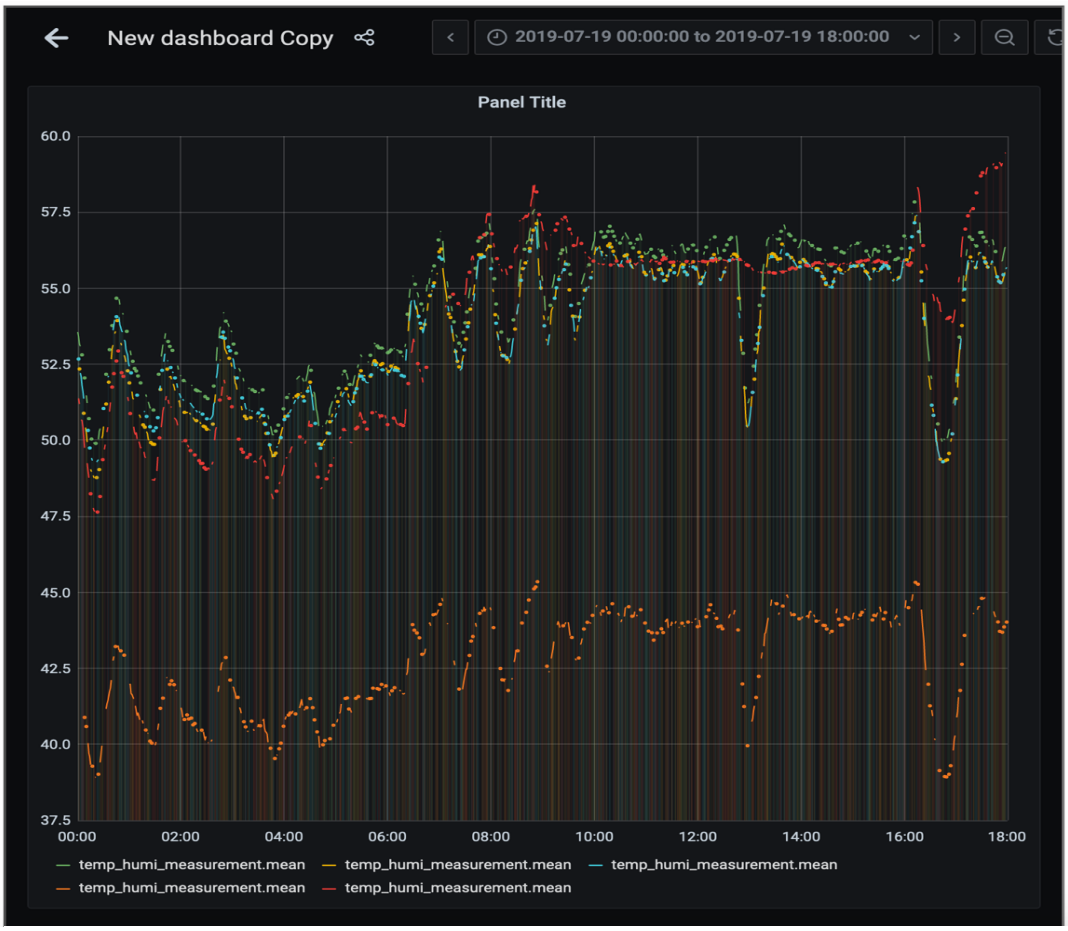

Figure 8 Visualization of Senselet environmental data in an iframe embedded Grafana dashboard

5. Evaluation

This section evaluates, analyzes and explains the working of the framework with the help of metrics.

5.1. Evaluation Set Up

The equipment used to run all experiments and tests is a 2018 15-inch MacBook Pro running macOS Big Sur Version 11.2.3. It is operating on an Intel Core i7 processor at 2.6 GHz. It has 16 GB 2400 MHz DDR4 RAM in addition to an Intel UHD Graphics 630 1536 MB card.

5.2. Metrics and Tables

Below are tables containing metrics from experiments run on the fusion framework on the evaluation set up mentioned above:
Scenario 1 is the testing of the iframe aspect of the framework and is done with two queries. The first query has a range of one day, while the second one has a 10-day range. This time metric measures the time taken to fetch and load the data visualization in the iframe. The data loaded consists of the data that is fetched and rendered onto the Grafana dashboard specified in the query.

Table 1 Metrics for Scenario 1

Time Taken to load the Grafana iframe	460 ms
Data loaded on initial iframe rendering	28.72 KB
Time taken to load dashboard	222 ms
Data loaded for dashboard for 30 min range	277 B
Data loaded for query #1 (1-day range)	109 KB
Data loaded for query #2 (10-day range)	189 KB

Scenario 2 tests the time taken to run a direct query to InfluxDB versus the time the extractor takes for the entire process. The first query has a range of 1 day, while the second one has a 10-day range. The command line time metric measures the time taken to fetch and return the data for the query as specified. Meanwhile, the 4CeeD Extractor runtime includes the time taken to call the extractor, pass the parameters, query InfluxDB, pre-process the data and convert it to a Python dictionary before uploading it to Clowder. As a result of these extra steps involved in the 4CeeD extractor, we see that the runtimes are also longer for identical queries.

Table 2 Metrics for Scenario 2

Command Line InfluxDB query runtime for 1 day	2.65 s
4CeeD Extractor runtime for one day	12.22 s
Command Line InfluxDB query runtime for 10 days	3.67 s
4CeeD Extractor runtime for 10 days	15.49 s

5.3. Analysis

The integration framework builds on the existing 4CeeD codebase that is based on Clowder. The software implementation for the page with the iframe is done in HTML and embeds a Grafana dashboard into 4CeeD. The Grafana dashboard visualizes the Senselet data consisting of environmental data such as humidity and temperature information stored in InfluxDB and is rendered in 460ms onto the webpage. 460ms is a short loading period since the initial dashboard is loaded only with recent data that is not data-heavy. Grafana comes with built-in support for InfluxDB and is therefore extremely efficient in visualizing the stored data. The size of the iframe subdocument rendered initially is 28.43KB and loads quickly without the need for extensive resources. We are also able to see how the browser needs to render and fetch more data for Query 2 since it has a longer duration. The metrics show us that the time taken by the extractor is significantly higher since there are multiple more phases of communication now. Additional time is taken to call the extractor, parse of the date range, process the time-series environmental data returned from InfluxDB and upload the metadata to Clowder. This difference in runtime is also caused as a result of network delays that occur between the extractor and InfluxDB instance as well as the extractor and Clowder instance during the data transfer process.

4CeeD/Clowder uses the Scala Play framework to integrate components and APIs for this web application. It also uses the Model-View-Controller (MVC) architecture that provides functional programming patterns for easy and quick development. The integration framework developed also ensures the safety and security of experiment data as it is only accessible to those with credentials to log into 4CeeD for their experiments. Thus, the software implementation ensures the swift and efficient rendering of dashboards visualizing Senselet data in a secure and coordinated manner. The integration framework itself does not use any hardware and is processed by the same webservers as 4CeeD. The metrics further confirm our fundamentals about the framework as they logically correspond to the technical aspects of each scenario.

5.4. Lessons Learned

The implementation of this integration framework was challenging due to a multitude of reasons. The two platforms, i.e., Senselet and 4CeeD, have very different infrastructures and designs. Senselet is used to store environmental data such as humidity and temperature information in a time-series format using InfluxDB. Meanwhile, 4CeeD uses MongoDB to store unstructured data from experiments in a completely different manner. The two platforms utilize different technologies and frameworks and are therefore dissimilar in many ways. In order for researchers to be able to take action as soon as possible, it is also essential that an integrative framework allows them to view and analyze environmental data from Senselet in real-time. In addition to this, it is also important for all access control and security policies to be maintained between the two platforms to ensure integrity and accuracy. All of these points make it challenging to develop an integrative framework that solves the researchers' problem in an efficient and maintainable fashion.

Senselet and 4CeeD are two platforms that are relatively new and not as widely adopted as some other data management systems used. This makes it challenging to develop new features swiftly as a result of fewer examples and documentation. As a result of fewer platform experts, the process of understanding the systems and developing custom features becomes more challenging. At the same time, it also essential to adopt less invasive approaches to ensure that the added feature does not disturb the older ones or add to the complexity of the platform. The design of such an integrative platform took a long period of time in order to ensure that all of the challenges were solved while developing an efficient,

lightweight and maintainable solution. The design process was thus challenging as it needed to keep a wide variety of factors in mind. In addition, it was also crucial for the researchers and scientists to be on the same page during the development of the design and features in order to develop a solution that solved the sneakernet problem between the Senselet and 4CeeD systems.

It was crucial to understand the systems and become comfortable with the respective codebases during the development process. I needed to study more into the documentation of each platform, understand the respective frameworks (e.g., Play Framework for 4CeeD) and create a development environment with all required technologies. It was also a challenge to run and test multiple technologies such as RabbitMQ, MongoDB, 4CeeD, and Grafana all at the same time while they interact with each other. Once I finished development, it was important for me to test all the features and ensure all use cases were satisfied. It was also crucial to test edge cases to ensure that the framework was stable for researchers to use and not buggy. Finally, I evaluated the platform, tested metrics such as runtime and data downloaded, and documented all the details. Throughout my research process, I learned a great deal and realized how crucial it is to keep in mind the best practices for software development. I received good feedback from researchers about the framework as they expressed their delight with the platform. They also suggested bonus features that could be added in the future such as additional plots and custom time ranges in the extractor.

6. Conclusion

4CeeD is an excellent cloud system that can help preserve data from experiments and benefit researchers for years to come. The ability to capture, store, analyze, and correlate digital data in real time, combined with high availability, is beneficial for researchers. Research reveals that the 4CeeD curation service will reduce the time spent at digital microscopes by almost a third and help alleviate data capacity and security concerns. 4CeeD can help researchers significantly save time and cost spent on experiments while effectively dealing with high-volume workloads.

The technical design of the system can handle fast-changing workloads with all sorts of experimental data. The choice of document-based MongoDB allows 4CeeD to be flexible when storing heterogeneous types of data. Groups have also noted that 4CeeD can help them conduct experiments 30% more quickly, leading to savings of over $30,000 per student throughout their Ph.D. when it comes to lab time. Thus, 4CeeD can help shorten the multi-decade-long development of new materials while providing security and convenience. Other systems such as Google Drive and DropBox are not built for scientific and research use cases, while NanoHub and DataUp focus more on making existing data more accessible. 4CeeD solves the problem at the bud by covering the end-to-end process from capture to analysis.

Senselet is another excellent system that uses IoT and sensor-edge cloud solutions to help control and monitor academic cleanrooms. The reliability and availability of data offered are beneficial for researchers. In addition, the emergency alerts in real-time through anomaly detection serve as a helpful check for cleanroom administrators. The use of cloud technology and IoT sensors to solve this problem is an excellent solution for academic environments.

This framework solves the critical issue researchers face where they cannot efficiently correlate scientific data between Senselet and 4CeeD. It removes the tedious and manual process scientists faced in analyzing and visualizing environmental data concerning their experiments. The project makes it easier for researchers to detect anomalies and notice trends in their data, helping them to speed up the process involved in discoveries. It also improves preventative maintenance as researchers can easily correlate their external and experiment data and make sense of any possible issues. They can better understand their experiments, diagnose issues and improve the development process. This framework uses the least intrusive practices to efficiently integrate the two frameworks while providing all

necessary features to researchers. It uses existing projects such as the dashboards in Grafana and therefore minimizes the additional work done. After considering all approaches, this project uses an efficient method to solve the problem faced by researchers while minimizing the effort required in software development and hardware upkeep.

References

[1] Timely and Trusted Curation and Coordination. (n.d.). Retrieved December 6, 2020, from https://t2c2.csl.illinois.edu/

[2] 4CeeD. (n.d.). Retrieved December 6, 2020, from https://4ceed.github.io/

[3] Grainger Engineering Office of Marketing and Communications. (n.d.). SENSELET Provides Sensory-Driven IoT Network for Scientific Instruments. Retrieved December 6, 2020, from https://cs.illinois.edu/news/senselet-provides-sensory-driven-iot-network-scientific-instruments

[4] Grainger Engineering Office of Marketing and Communications. (n.d.). SENSELET provides sensory-driven IoT network for scientific instruments. Retrieved December 6, 2020, from https://mntl.illinois.edu/news/10977

[5] System Properties Comparison InfluxDB vs. MongoDB. (n.d.). Retrieved December 6, 2020, from https://db-engines.com/en/system/InfluxDB;MongoDB

[6] Churilo, C. (2020, March 24). MongoDB vs InfluxDB: InfluxData Time Series Workloads. Retrieved December 6, 2020, from https://www.influxdata.com/blog/influxdb-is-27x-faster-vs-mongodb-for-time-series-workloads/

[7] Nguyen, Phuong, Konstanty, S., Nicholson, T., Brien, T., Schwartz-Duval, A., Spila, T., Nahrstedt, K., Campbell, R., Gupta, I., Chan, M., McHenry, K., Paquin, N. (2016). 4CeeD: Real-Time Data Acquisition and Analysis Framework for Material-related Cyber-Physical Environments. 10.13140/RG.2.2.31064.08969.

[8] Nahrstedt, K., Yang, Z., Yu, T., Su, P., Kaufman, R., Shan, I., ... & Dallesasse, J. (2020). Senselet: Distributed Sensing Infrastructure for Improving Process Control and Safety in Academic Cleanroom Environments. GetMobile: Mobile Computing and Communications, 24(2), 12-16.

[9] Welcome to the Coordinated Science Lab at Illinois homepage! (n.d.). Retrieved December 6, 2020, from http://www.csl.illinois.edu/news/illinois-researchers-build-dropbox-storage-analytical-system-scientific-data

[10] Clowder framework. (n.d.). Retrieved April 18, 2021, from https://clowderframework.org/

[11] Html tag. (n.d.). Retrieved April 18, 2021, from https://www.w3schools.com/tags/tag_iframe.asp

[12] Web technology for developers. (n.d.). Retrieved April 18, 2021, from https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe

[13] Grafana. (2021, April 9). Retrieved April 18, 2021, from https://en.wikipedia.org/wiki/Grafana

[14] Real Python. (2021, March 6). Python's requests Library (guide). Retrieved April 18, 2021, from https://realpython.com/python-requests/

[15] Clowder-Framework. (n.d.). Clowder-framework/pyclowder. Retrieved April 18, 2021, from https://github.com/clowder-framework/pyclowder

[16] Python (n.d.). Retrieved April 18, 2021, from https://influxdb-python.readthedocs.io/en/latest/include-readme.html

[17]Marini, L., Gutierrez-Polo, I., Kooper, R., Satheesan, S. P., Burnette, M., Lee, J., . . . McHenry, K. (2018). Clowder. Proceedings of the Practice and Experience on Advanced Research Computing. doi:10.1145/3219104.3219159