Project acronym	B4B
Project full name	Brains for Building's Energy Systems
Grant No	MOOI32004
Project duration	4 year (Starting date May 1, 2021)

Deliverable D4.02
Data test set for internal research purposes
Joep van der Weijden (TUDO, Mirjam Harmelink (TUD), Pieter Pauwels (TU/e)

Work package	Work Package 4
Result	8
Lead beneficiary	TU Delft
Due Date	30 April 2022
Deliverable Status	FINAL
File name	B4B-WP4-T4.02- Compilation of data sets for internal research purposes FINAL.docx
Reviewers	Olivia Guerra Santin (TU/e)

The project was carried out with a Top Sector Energy subsidy
from the Ministry of Economic Affairs and Climate, carried out
by the Netherlands Enterprise Agency. The specific subsidy for

SAMENVATTING

Het doel van het Brain4Buildings project is het toevoegen van operationele intelligentie aan gebouwen om de transitie naar energie-efficiëntie en –flexibiliteit te bewerkstelligen. Om algoritmen te ontwikkelen welke operationele intelligentie toevoegen zijn datasets van bestaande gebouwen nodig. Verschillende consortium partners stellen gebouw data beschikbaar, wat optelt tot datasets van 17 verschillende gebouwen. Deze datasets zijn verdeeld in Living Labs, Use-Cases and Validation Cases.

Dit document geeft een overzicht van de beschikbare datasets van Living Labs. Daarnaast worden de keuzes en vereisten voor dataopslag in een consortium van 39 partijen besproken. Verschillende typen data, zoals historische datasets versus live data stromen, vereisen verschillende typen platformen. Ten slotte beschrijft dit document het proces om toegang te krijgen tot de verschillende platformen, inclusief het gebruik van een dat uitwisselingsovereenkomst en datamanagement plan.

SUMMARY

The goal of the Brains4Buildings consortium is to add operational intelligence to buildings to facilitate the transition towards more energy-efficient and flexible buildings. To develop algorithms to achieve operational intelligence, datasets from existing buildings are required. Several consortium partners are sharing building data, which adds up to datasets from 17 different buildings. These datasets are divided into three groups: Living Labs, Use Cases and Validation Cases.

This document gives an overview of the available data from the Living Lab datasets. Furthermore, the choices and requirements regarding data storage in a consortium of 39 parties is discussed. Diverse types of data, like historical datasets versus live data streams, require distinct types of data platforms. Finally, this document describes the process of getting access to the different data platforms, including the use of a data exchange agreement and data management plans.

TABLE OF CONTENT

TABLE OF FIGURES

1 INTRODUCTION

Brains4Buildings is a consortium of several parties, including companies and knowledge institutes. The goal of this consortium is to add operational intelligence to buildings to facilitate the transition towards more energy-efficient and flexible buildings.

Work package 4 elaborates on data connectivity between data sources in buildings, the security of data, the ethical implementation of data sharing, and standardization. Research within the B4B project focuses on data integration on API-level instead of the data level of individual systems from different manufacturers. To investigate how data can be integrated on the API level we collect and analyze data from several buildings.

The Brains4Buildings consortium is currently (April 2022) investigating the potential use of data from 17 different buildings from several consortium partners. A distinction is made between Living labs, Use cases and Validation cases.

- Living labs are buildings that offer complete freedom and are already the subject of years of research and much is known.
- Use cases are buildings where parties have already started collecting data, much is known but much less historical data and fewer opportunities for experimentation.
- Validation cases are buildings for which no data has yet been collected, but which should start with the intention that we can use them to test one or more developed applications.

For some of these 17 buildings, a lot of information can be shared, and access is possible to data from the Building Management System (BMS), Energy Management System (EMS) and/or the smart energy meter. For other buildings, the data cannot be shared due to the lack of digital maturity in buildings, or the security of the intellectual property of the data. Therefore, some data is shared between all consortium partners, while other datasets are kept in relative isolation and only shared with specific researchers.

Secure data storage and sharing within organizations are often well taken care of by the ICT department of the individual organization, but securely sharing larger data sets between organizations is an issue. Emailing Excel files or sharing through services like Dropbox or WeTransfer is not secure or scalable. This document proposes solutions for data storage and sharing within the Brains4Buildings consortium. It is discussed which type of data is available to the consortium, where several types of data are stored and how to get access to several types of information and datasets.

2 DATA STORAGE AND SHARING FACILITIES

2.1 Introduction

Several factors determine which data storage and sharing service are ideal for the desired purpose. The most deciding factors are 1) the volume of data, 2) whether the data is to be shared between organizations, 3) if the data need to be made available to the entire consortium or not, and 4) if it is real-time data or historical data that need to be shared. Building data shared between the B4B consortium partners has a certain level of sensitivity. Not because privacy-sensitive data is collected and shared, since this type of information and data is easily avoided in utility buildings, but due to intellectual property rights of the building data. The used storage and sharing platform therefore should allow restrictive access as safe data sharing between organizations needs to be a possibility.

As mentioned in chapter 1, some consortium partners are sharing data and building information with specific researchers using their own procedures. This is mainly the case for the buildings used as use-cases or validation cases. For the living labs, the data is shared within the consortium.

Figure 1 shows the timeline and which type of storage platforms will be used for which type of data in these living labs. These different platforms are discussed below.

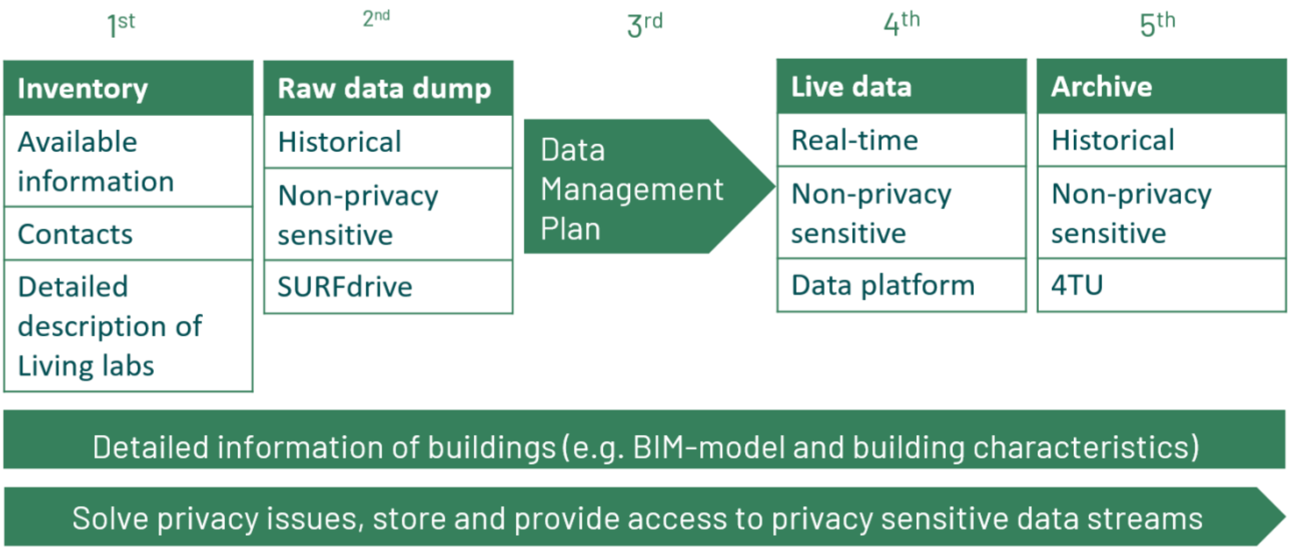

Figure 1 Steps for data collection and storage within the B4B project

2.2 SURFdrive for historical datasets and building information

Since there is a need to start sharing historical datasets within the consortium as soon as possible, we chose to start with a data dump on a cloud-based drive. Most people are familiar with Dropbox and Google Drive, but these services are not supported by all organizations. Microsoft provides shared services through Sharepoint, but the drive has a limit of 1GB of storage. For the B4B project we decided to use SURFdrive. It has similar functionalities as the beforementioned platforms. There is, however, an important added benefit of SURF it being a non-commercial company with a focus on education and research. SURFdrive complies with all Dutch and European privacy legislation. The data are stored safely in the Netherlands and are never made available to third parties. The location of the data is always known.

SURFdrive will be used for two distinct types of data:

- A data dump with historical BMS data, sensor data, energy consumption, etc, and
- Background information about the buildings like building characteristics and principal schemes.

Access to the files on SURFdrive can be customized per user. There is the possibility to give the following rights:

- Viewing rights
- Viewing and uploading rights
- Viewing and editing rights

2.3 Research Data Platform for real-time data

Real-time data requires a different type of platform. This type of data is usually provided by a Building Management System (BMS), Energy Management System (EMS) and/or a smart energy meter. These types of systems are closed environments and only allow data to be extracted when the user has the right credentials. Data extraction is performed by making use of an Application Programming Interface (API). Buildings do not all use the same BMS or EMS system and therefore the APIs are all a bit differently structured. If researchers like to access the data from different buildings, they would all need to get different access credentials to each of the BMS or EMS systems and learn how each of the APIs is structured. This requires a relatively high level of programming skills and takes a lot of time.

In the past years, The Green Village and SURF have developed a service focused on live data streams. This service is called the Research Data Platform. Information about this platform can be found on the GitLab page. By using this platform to connect to the APIs of the BMS systems of the buildings in Brains4Buildings, the real-time data streams from different buildings become accessible in a single location. In this way, the different researchers in Brains4Buildings do not need to learn all these different APIs to access data streams for different buildings. Instead, the connection to these BMS APIs is made one time by a programmer, after which the data is easily accessible through the Research Data Platform. To reduce the risks involved with storing sensitive data, the real-time data is only stored on the Research Data Platform for 7 days.

The data on the Research Data Platform is obtained by writing a Kafka Consumer. An example of a script for a Consumer is presented in Appendix 3, which can be used as a template. When access is requested, according to the workflow presented in Chapter 4, a, API key and API secret are supplied. With these credentials, it becomes possible to get access to the database with live data.

2.4 4TU Research Data for archiving real-time data

4TU Research Data, established in 2008, is an international data repository for science, engineering, and design. It provides long-term access and preservation of research datasets and other services like curation and sharing. Furthermore, 4TU Research Data offers training and resources to researchers to support them in making research data findable, accessible, interoperable, and reproducible (FAIR).

Since the real-time data streams are only allowed to be stored on the Research Data Platform for 7 days, there needs to be some form of archiving of this data. To archive the real-time data streams from the Research Data Platform and build a historical dataset from these streams, it is planned to make a connection between the Research Data Platform and 4TU Research Data. As of this moment (April 2022) this connection has not been made.

2.5 Other data sharing methods

As mentioned in the introduction, the data from the living labs are shared through the platforms mentioned above. However, the data from use cases and validation cases is not necessarily shared through these means. E.g., TNO, Cloud Energy Optimizer, DWA and BAM mentioned that their building data is not open data, and will be shared directly with specified researchers on request

3 OVERVIEW OF BUILDINGS AND AVAILABLE DATA SETS

The Brains4Buildings consortium is currently (April 2022) investigating the potential use of data from 17 different buildings from several consortium partners. A distinction is made between Living labs, Use cases and Validation cases.

- **Living labs** are buildings that offer complete freedom and are already the subject of years of other research and much is known.
- **Use cases** are buildings where parties have already started collecting data, much is known but much less historical data and fewer opportunities for experimentation.
- **Validation cases** are buildings for which no data has yet been collected, but which should start with the intention that we can use them to test one or more developed applications.

The Living lab buildings (April 2022) are:

- Kropman Breda
- HHS: Education building
- TU Delft: Wiskunde & Informatica
- TU Eindhoven: Atlas Living Lab

The other buildings (Use cases and Validation cases) of which data is used in the consortium (April 2022) are:

- TU Delft: Pulse
- TU Delft: Aula
- Avans, Breda
- Hogeschool Windesheim: Building A
- Hogeschool Windesheim: Building X
- TU Delft CiTG building
- TNO Stieltjesweg
- DWA Office
- Strukton Science Park
- Cloud Energy Optimizer
- RHDHV Amersfoort
- Kuijpers Den Bosch
- Bam Bunnik

The first research will be performed on the data from the current Living labs. Therefore, the focus is to make the data from these buildings available as quickly as possible. An overview of the buildings used as Living Labs and their key characteristics are presented in Table 1. More detailed information can be found on Sharepoint[1].

[1] https://tud365.sharepoint.com/:f:/r/sites/KernteamB4B/Gedeelde%20documenten/General/07_Living%20labs/03_Results%20per%20building?csf=1&web=1&e=U7o5cB

Table 1 Overview of key characteristics of Living Labs (April 2022)

	Building	Kropman Breda	HHS: Education building	TU Delft: Wiskunde & Informatica (building #28)	TU Eindhoven: Atlas Living Lab
Building characteristics	Year built (renovation)	1993	2009	2002 (2018)	Renovated in 2018/2019
	Area (m2)	1500	16270	10.767	44.000
	Energy label	No label	Label A. EPC=0,329	A (label in 2009 before the renovation)	A++++
	Renewable production	PV Panels	Heat pump, ATES system, solar panels, solar collector system for domestic hot water	No	ATES, geothermal energy system
	BMS system	InsiteSuite	Priva	Johnson Control	Honeywell Building Technologies NL Honeywell EBI500
	Energy systems (main components)	Heating: Gas fired HR-boiler Cooling - Electric, air source Air handling unit (AHU) Central AHU Storage: Battery storage 57kWh	ATES system, heat pump, gas boiler	Two high efficiency gas boilers	ATES / geothermal energy system / Heat pumps
Data availability	BMS data	(Historical) BMS data are available for 18 years BMS data can be approached in real time	(Historical) BMS data are available	(Historical) BMS data are available for 2019, 2020 and 2021 Realtime BMS data through Data platform of TU Delft	(Historical) BMS data are available for 2019 and 2020 BMS data can be approached in real time
	BMS naming convention	InsiteSuite	Priva	Haystack	Specific to building. A mapping table is available.
	Energy meter data	(Historical) Energy meter data are available for 4 years Energy meter data can be approach in real time	(Historical) Energy meter data are available	(Historical) Energy meter data are available for multiple years Energy meter data can be approached in real time through Data platform of TU Delft	(Historical) Energy meter data are available for 2019 and 2020 Energy meter data can be approached in real time
	Occupancy behavior data	No data collected		WiFi data, pseudonymized	Thermostat control point adjustment of +/- 2°C in rooms. Hourly data
	BIM data	No BIM model ACAD drawings			Yes, including Architectural (Floor plans, sections, elevations), Architectural façade, Construction, Water pipes, sprinklers, Sanitary installations,

					Climate ceilings, Air ducts

Currently (April 2022), not all data from living labs are accessible for research. For each of the living labs, the general building description of the type of systems in the building is available. These can be found on Sharepoint[2]. For Building 28 of the TU Delft, both the data dump, as well as the live data streams are available. An overview of the availability of data or contextual information of the buildings can be found in Table 2.

 Available

 Work in progress

 Not available through SURFdrive or Research Data Platform

Table 2 Overview of available data and contextual information of Living Lab buildings (April 2022)

	Kropman Breda	HHS: Education building	TU Delft: Wiskunde & Informatica (building #28)	TU Eindhoven: Atlas Living Lab
General building description				
Building information				
Historical data dump				
Live data streams				

2 https://tud365.sharepoint.com/:f:/r/sites/KernteamB4B/Gedeelde%20documenten/General/07_Living%20labs/03_Results%20per%20building?csf=1&web=1&e=U7o5cB

4 REQUESTING ACCESS TO DATASETS

As mentioned previously, data from most of the living labs are shared within the consortium, while other data streams are shared with specific researchers. Access to these data can be requested by reaching out to the consortium partners who own these data.

4.1 Access to data from open living labs

The datasets include sensitive information, which is why the data is not made public. The sensitivity is not due to privacy but relates to the intellectual property of the data. Therefore, the datasets need to be protected from data breaches.

To protect the datasets from data breaches, authorization and authentication will take place. Authentication is the process of verifying who someone is, whereas authorization is the process of verifying what specific applications, files, and data a user has access to. Next to the verification, partners need to agree that they will handle the used data securely. To formalize this part of the process, a Data Exchange Agreement is constructed. A template for this document is currently in construction and will be supplied as soon as it is finished.

Since the people in Work Package 4 do not know everyone in the consortium, this process is streamlined by assigning the authentication role to the work package leaders. They will verify that the request for data access is legit and forward the request to the data management team, primarily Joep van der Weijden (j.a.m.vanderweijden@tudelft.nl). This process is shown in Figure 2.

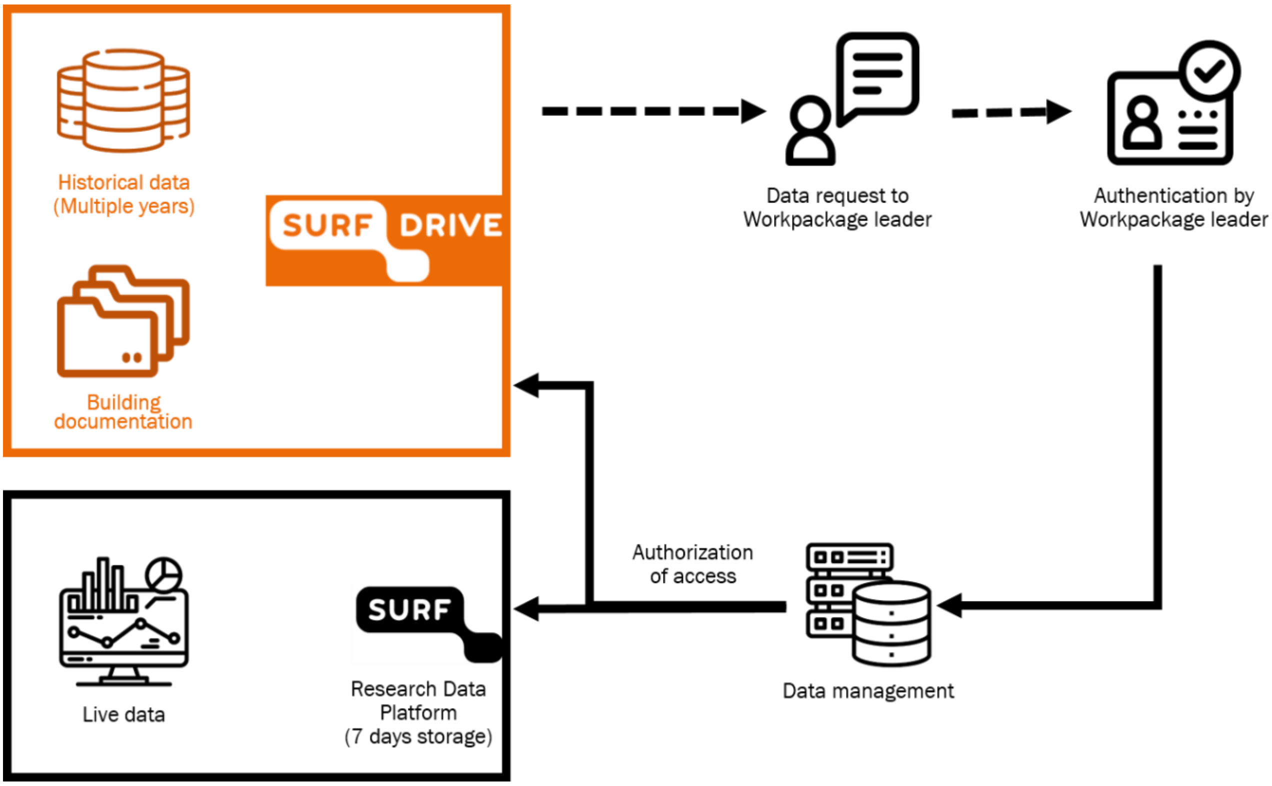

Figure 2 Process for data access

4.2 Access to data from other buildings

Access to the data which is not shared through SURFdrive and/or the Research Data Platform can be requested by addressing the consortium partner involved with that building. An overview of the contact person per building can be found here. If needed, the consortium partner can alter the Data Exchange Agreement template to make sure that the data is handled as promised.

4.3 Data Management plan (DMP)

A data management plan is a document that describes how the data will be generated or used within a given project. In a data management plan, the researcher describes how data will be collected, managed, stored and made available during the research period, and how they will be shared upon completion of the research project.

A data management plan helps to determine how the data can be managed efficiently, effectively, and securely. In addition, planning for good data management from the start reduces the risk of data loss, data breach, or other threats that could render the data illegible or unusable (e.g., the obsolescence of software).

A data management plan is mandatory for researchers of the TUD and TUe, and will need to be shared in order to gain access to the data streams within Brains4Buildings. Sample DMP's for the TU/e and TUD is presented in Appendix 1 and Appendix 2.

APPENDIX 1: DATA MANAGEMENT PLAN TU/E

Project Name

Brains for Buildings: Data Integration for Smart Buildings

Project description

This project is a part of Brains for Building's Energy Systems (B4B) project. The project is focused on developing methods to integrate data available from smart meters, building management systems and the Internet of Things devices, to develop methods to reduce energy consumption, increase comfort, respond flexibly to user behaviour and local energy supply and demand, and save on installation maintenance costs. This will be done through the development of faster and more efficient Machine Learning and Artificial Intelligence models and algorithms. The project is geared to existing utility buildings such as commercial and institutional buildings.

Data Management Plan

1	**General information**	
1.1	Project title	Brains for Buildings: Data Integration for Smart Buildings
1.2	Name and affiliation of principal researcher	XXX, PhD Candidate, Department of Built Environment, Eindhoven University of Technology
1.3	List all users of the data, including name, affiliation, and role.	XXX, PhD Candidate, Department of Built Environment, Eindhoven University of Technology
2	**Description of datasets**	
2.1	Describe the purpose for which the dataset is to be used.	Historical data sets will be required for the purposes described below. For developing data integration methods to integrate various data silos in the built environment such as Building Management Systems data, Energy Management Systems and Building Information Models. Use the integration platform to develop methods to reduce energy consumption, increase comfort, respond flexibly to user behaviour and local energy supply and demand, and save on installation maintenance costs.
2.2	Provide a general description of the type of data you will be working with, including any re-used data	Following data sets related to the Atlas Building, TU Eindhoven campus. Data from Honeywell BMS indoor environmental quality parameters (illumination, CO2, ppm, humidity, temperature, VOC, CO, air velocity), HVAC machinery characteristics like refrigerant temperature, electrical current, voltage, power factor, vibration, return and supply air temperature, set points, fault logs Data from Envision Manager (Signify) Energy, occupancy, temperature, light levels Data from GMC Energy Management energy consumption (water, gas, electricity) BIM models with architectural, electrical, mechanical, plumbing systems
2.3	Which type and format of data are the data?	BMS data in .csv, spreadsheets, .json BIM models in .rvt
2.4	Which data types are sensitive and why?	Occupancy data is sensitive if it contains data that can be used to identify a person directly or indirectly
2.5	Motivate the level of detail of the data that is needed for your purpose. (E.g., raw data or aggregated data over time/location/etc.)	BMS data: raw data with highest granularity possible for time and metering levels. Different analysis may have different requirements. GMC Energy Management: raw data with highest granularity possible for time and metering levels. Different analysis may have different requirements.

		Envision Manager (Signify): raw data with highest granularity possible for time and metering levels. Different analysis may have different requirements. BIM models: BIM model including architectural, mechanical, electrical, plumbing and all sensors and devices
2.6	What documentation will accompany data?	Data dictionary explaining the variables used, documentation explaining how data is organised
2.7	State the time/duration that the dataset spans	Data from Honeywell BMS 2 years Data from Envision Manager (Signify) From 2021 onwards Data from GMC Energy Management 2 years
3	**Data storage and processing during and after the research**	
3.1	Where will the data be stored during the research? List all storage locations in case there are multiple	Personal laptop Personal Tue OneDrive
3.2	What is the expected volume of the data?	250 GB
3.3	Where will the data be stored after the research? List all storage locations in case there are multiple.	Data will only be stored within the period of project until 30 April 2025
3.4	In case you plan to combine the data with one or multiple existing datasets, describe these other dataset(s) and the privacy implications of this processing.	Sensitive personal information is not used
3.5	Are there expected costs? Please specify these and state an amount that is as realistic as possible. How will the costs be covered?	No anticipated costs
4	**Legal and ethical requirements, codes of conduct**	
4.1	Does the research involve human subjects or 3rd party datasets collected from human participants?	No
4.2	Does the research work with personal data? (Information about an identified or identifiable natural person)	No
4.3	Which personal data will you process?	None
4.4	Does the research work with any types of confidential or classified data or code?	No
4.5	How will ownership of the data and intellectual	No personal data is collected

	property rights to the data be managed?	
4.6	What is the legal ground for personal data processing?	No personal data is collected
4.7	Please describe the informed consent procedure you will follow	No human subjects are used and therefore no consent form is required
4.8	Where will you store the signed consent forms?	No
4.9	Does the processing of the personal data result in a high risk to the data subjects?	No personal data is collected or processed
4.10	What will happen with personal research data after the end of the research project?	No personal data is collected or processed
5	**Standards and Metadata**	
5.1	Will a standard be used for the metadata? If yes, describe in detail which, and state in which databases these will be included. If no, state in detail which metadata will be made to make the data easy/easier to trace and make available for reuse. Mention the database in which these metadata will be included.	No
6	**Sharing of data**	
6.1	For which researchers/research groups will the personal data be made available?	No personal data is collected
6.2	For which researchers/research groups will the other data be made available?	For the principle researcher
6.3	State what the current plan for publication is relating to the dataset.	Raw data or data in any way that allows identification of persons, directly or by combining the data with other data sources is not permitted will not be published.

Signatures

Name : XXX XXX
Affiliation : PhD Candidate
Date : 26/10/2021
Function : Principal researcher

Name : XXX XXX
Affiliation : Associate Professor
Date : 26/10/2021
Function : Supervisor

Name : XXX XXX
Affiliation : Assistant Professor
Date : 26/10/2021
Function : Co-Supervisor

APPENDIX 2: DATA MANAGEMENT PLAN TUD

APPENDIX 3: EXAMPLE KAFKA CONSUMER CODE

```
from confluent_kafka import Consumer, KafkaError, KafkaException
from confluent_kafka.schema_registry import SchemaRegistryClient
from confluent_kafka.schema_registry.avro import AvroDeserializer
from confluent_kafka.schema_registry.error import SchemaRegistryError
from confluent_kafka.serialization import SerializationContext, MessageField
from confluent_kafka.error import SerializationError

import json
import os
import requests
from pprint import pformat


if __name__ == '__main__':

    bootstrap_servers = os.environ["BOOTSTRAP_SERVERS"]
    api_key = os.environ["API_KEY"]
    api_secret = os.environ["API_SECRET"]
    topic = os.environ["TOPIC"]
    schema_registry_url = os.environ["SCHEMA_REGISTRY_URL"]
    schema_registry_username = os.environ["SCHEMA_REGISTRY_USERNAME"]
    schema_registry_password = os.environ["SCHEMA_REGISTRY_PASSWORD"]
    consumer_group = os.environ["CONSUMER_GROUP"]

    # SchemaRegistryClient will be needed to instantiate AvroDeserializer.
    conf = {
        'url': schema_registry_url,
        'basic.auth.user.info':
f"{schema_registry_username}:{schema_registry_password}"
    }
    schema_registry_client = SchemaRegistryClient(conf)

    # Get the schema from the Schema Registry.
    r = requests.get(f"{schema_registry_url}/subjects/{topic}-
value/versions/latest",
                     auth=(schema_registry_username, schema_registry_password))
    print(r.status_code)
    schema_str = r.json()['schema']
    print(schema_str)

    # AvroDeserializer and SerializationContext will be needed to deserialize
    # individual messages when a regular Consumer is used.
    # (DeserializingConsumer takes care of the deserialization internally.)
    deserializer = AvroDeserializer(schema_str, schema_registry_client)
    ser_context = SerializationContext(topic, MessageField.VALUE)

    # Consumer configuration
    # See https://github.com/edenhill/librdkafka/blob/master/CONFIGURATION.md
    conf = {
        # Choose a unique consumer group id and a client id.
        'group.id': consumer_group,
        'client.id': 'test-consumer',
        'bootstrap.servers': bootstrap_servers,
        'sasl.mechanisms': 'PLAIN',
        'security.protocol': 'SASL_SSL',
        'sasl.username': api_key,
```

```
        'sasl.password': api_secret,
        # 'ssl.ca.location': '/etc/ssl/certs/ca-certificates.crt',
        # 'auto.offset.reset': 'earliest',
        'auto.offset.reset': 'latest',
        'enable.auto.commit': 'false'
    }

    # Create a Consumer instance.
    consumer = Consumer(conf)

    def print_assignment(consumer, partitions):
        print(f'Assignment: {partitions}')

    # Subscribe to the topic.
    consumer.subscribe([topic], on_assign=print_assignment)

    # Read messages from Kafka.
    try:
        while True:
            try:
                # Instead of using poll that returns messages one-by-one,
                # we use consume to receive a list of messages.
                # In this case, we receive a list of messages every time
                # there are 1000 new messages or 1 second elapses.
                msgs = consumer.consume(num_messages=1000, timeout=1.0)
                print(f"Received {len(msgs)} messages.")

            except KafkaError as e:
                print("KafkaError occurred, no offset will be committed:
{}".format(e))
                continue

            # Loop over the batch of messages obtained from Kafka.
            # Individual messages will be deserialized.
            # Messages that cannot be deserialized will be ignored.
            # In case any other errors from Kafka occur, no offset will be commited
to Kafka.
            msg_err = False
            values = []
            for msg in msgs:

                if msg.error():
                    # On error, break and do not commit any offset to Kafka.
                    msg_err = True
                    raise KafkaException(msg.error())
                    break

                print('%s [%d] at offset %d' % (msg.topic(), msg.partition(),
msg.offset()))

                try:
                    value = deserializer(msg.value(), ser_context)
                    values.append(value)

                except SerializationError as e:
                    # Ignore the messages that cannot be deserialized.
                    print("SerializationError occurred, the message will be ignored
and the offset will be committed: {}".format(e))
```

```
            continue

        # On error, do not commit any offset to Kafka and try again.
        if msg_err:
            continue

        # Process the whole batch of messages here at once.
        print(pformat(values))

        # Commit the last offset to Kafka on success.
        consumer.commit()

except KeyboardInterrupt:
    print('%% Aborted by user\n')

finally:
    # Close down consumer to commit final offsets.
    consumer.close()
```