D2.1

Extreme Data Flow Patterns – Report

IT4I@VSB

Public

Funded by
the European Union

Deliverable Properties

Version	v1.0
Dissemination level	PU
Deliverable type	R
Work package	WP2
Task	T2.1
Due date	30-09-2023
Submission date	30-09-2023
Deliverable lead	IT4I@VSB
Authors Name	Martin Golasowski (IT4I@VSB), Mohamad Hayek (BADW-LRZ), David Číž (IT4I@VSB), David Hurych (VALEO), Jan Zahradník (VALEO), Piyush Harsh (TERRAVIEW), Loïck Serbetot (ALTRNATIV)
Reviewers	Pinar Karagoz (METU), Jérôme Freani (ALTRNATIV)

Abstract

This deliverable of the EXA4MIND project collects and analyses data flow patterns from all the project application cases. The collected data flow descriptions are used to identify a set of common occurring patterns that will be taken into account when designing the Extreme Data Database.

Keywords

data flow patterns; extreme data; application cases; EXA4MIND; LEXIS

Document Revision History

Version	Date	Description of Change	Contributor(s)
v0.1	05-07-2023	Table of content	Mohamad Hayek (BADW-LRZ)
v0.2	30-08-2023	Inputs collected	Mohamad Hayek (BADW-LRZ), Martin Golasowski (IT4I@VSB)
v0.3	12-09-2023	Document ToC filled based on inputs from WP4, WP5, and WP6	Martin Golasowski (IT4I@VSB)
v0.4	15-09-2023	Identified patterns description added	Martin Golasowski (IT4I@VSB)
v0.5	18-09-2023	Text formatting and unification	Martin Golasowski (IT4I@VSB)
v0.5a	19-09-2023	Modification in common patterns section	Stephan Hachinger (LRZ)
v0.6	26-09-2023	Modifications based on feedback from reviewers and proofreading	Martin Golasowski (IT4I@VSB)
v1.0	28-09-2023	Final check by the coordinator	Kateřina Slaninová, Jan Martinovič (IT4I@VSB)

Disclaimer

Information, documentation and Figures available in this deliverable are provided by the EXA4MIND project's consortium funded by a European Union's Horizon Europe Research and Innovation programme under grant agreement No **101092944**. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Copyright Notice

Dissemination Levels

PU	Public	Public, fully open, e.g. website
SEN	Sensitive	Confidential to EXA4MIND project and Commission Services

Deliverable Types

R	Document, report (excluding periodic and final reports)
DEM	Demonstrator, pilot, prototype, plan designs
DEC	Websites, patent filings, press and media actions, videos, etc.
OTHER	Software, technical diagrams, etc.

Table of Contents

List of Figures

List of Tables

No tables are given.

Abbreviations

ADAS	Advanced Driver Assistance System
AI	Artificial Intelligence
AoI	Area of Interest
API	Application Programming Interface
AQIS	Advanced Querying and Indexing System
BPMN	Business Process Model and Notation
CNN	Convolutional Neural Network
DB	Database
DBMS	Database Management System
DoA	Description of Action
ECMWF	European Centre for Medium-range Weather Forecasts
EDD	Extreme Data Database
ETL	Extract, Transform, Load
GIS	Geospatial Information Systems
HPC	High Performance Computing
KPI	Key Performance Indentifier
LiDAR	Light Detection and Ranging
M2M	Machine-to-machine
MD	Molecular Dynamics
ML	Machine Learning
NFS	Network File System
OCR	Optical Character Recognition
SMC	Soil Moisture Content
SME	Small and Medium-sized Enterprises
UI	User Interface
USGS	United States Geological Survey
WB	Water Body
WP	Work Package

Table of Partners

Short Name	Partner
IT4I@VSB	IT4Innovations at VSB – Technical University of Ostrava (IT4Innovations) https://www.it4i.cz/en
BADW-LRZ	Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities https://www.lrz.de/english/
METU	Middle East Technical University (METU) https://www.metu.edu.tr/
AUSTRALO	AUSTRALO Interinnov Marketing Lab https://www.australo.org/
EURAXENT	Euraxent https://www.linkedin.com/company/euraxent/
CVUT	Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University (CIIRC CTU) https://www.ciirc.cvut.cz/
VALEO	Valeo https://www.valeo.com/en/valeo-ai/
TERRAVIEW	Terraview https://www.terraview.co/
ALTRNATIV	Altrnativ https://altrnativ.com/?lang=en

Executive Summary

This deliverable (due in M09) is the first of the WP2 deliverables and collects another important set of requirements needed for the design and implementation of the Extreme Data Database (EDD). Its results are used as valuable input for the WP1-driven co-design process of the EXA4MIND solution and also to drive the development of the EDD according to the application case requirements and needs.

1 Introduction

In this deliverable, we present the results of the initial data flow analysis performed by all the application cases in the EXA4MIND project. The main motivation for collecting this information is identification of common data flow patterns which will then be used as input for the Extreme Data Database (EDD) design and implementation.

This document also complements the Deliverable D1.1 - Application Cases and Architecture Requirements (EXA4MIND 2023), which contains the first set of detailed requirements on the EXA4MIND solution provided by the application cases and also technical partners of the project.

The same method used for initial identification of the data flows will be used continuously within the design process iterations.

As a formal way of describing the data flows, we use an approach similar to the one defined in the Business Process Model and Notation (BPMN) (Chinosi and Trombetta 2012). Each application case section is structured in the following way.

A brief introduction of the target domain is provided at the beginning, followed by a graphical diagram of the data flows and sections describing entities, data stores, flows and processes shown there.

The formal description is followed by a section focused on computational aspects of each application case, evaluating which processes provide the bulk of the compute workload and the related amount of data flow. A key terms glossary is provided by each application case to provide a clear definition of certain terms that can appear ambiguous within the scope of the entire project.

The data flows collected from each application case are presented in Section 2. The following Section 3 describes the initial identified patterns based on the provided data flows. The last Section 4 provides a summary and outlook of the next steps.

2 Initial Data Flows in EXA4MIND

Each subsection here provides a unified description of initial data flows for each application case. The data flow diagrams loosely follow the BPMN notation and for each case, a list of entities, processes, and flows is provided.

2.1 Scientific Application Case - ADAMS4SIMS (WP4)

This application case aims to develop an innovative automated data mining system for the systematic improvement of molecular simulation accuracy and predictability (ADAMS4SIMS). This section contains a description of the initial data flow for the Scientific Application Case (WP4) coming from the molecular dynamics simulation domain.

The appication case expects to build its own graphical user interface (UI) on top of the provided APIs of the EXA4MIND/LEXIS Platform services (LEXIS 2023). The data flows described here provide a top-level view of the integrated solution, including the custom-built UI. Due to complexity of this particular application case, we include

an additional Section 2.1.5 where the most important workflows which involve the individual data flows are described.

Figure 1 shows the data flow in the proposed ADAMS4SIMS system. It contains 1 external entity, 2 data stores, and 7 processes. These are described below in the following sections.

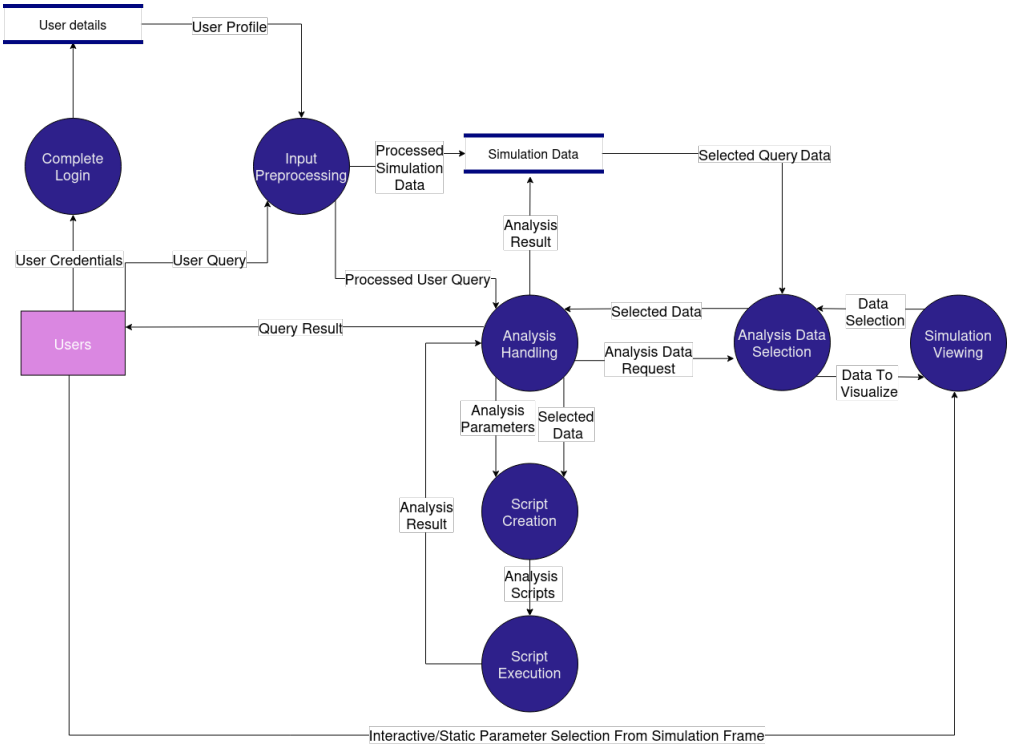

Figure 1: WP4 Data Flow Diagram

2.1.1 Entities

This application case has one type of an external entity, which is the Users, who interact with the system by creating new analyses and downloading/uploading new data. They have varying levels of privileges that denote what kind of queries they can run. For example, some users may only be able to view the simulation data, while others may be able to run complex analysis workflows on it.

Users pass User Credentials data to a *Complete Login* process for authentication and User Query data to *Input Preprocessing* for specifying the actions they want to do in the system.

User Query can be a request of an upload/download of data (simulation data, experimental data, analysis results) or running analysis workflows. For example, a user may want to upload some experimental data and compare them with the simulation data, or download the results of a previous analysis workflow. Alternatively, a user may want to run a new analysis workflow on a selected subset of the simulation data.

2.1.2 Data Stores

Two data stores are considered:

- **User Details** – holds User Profile data and passes them along to the *Input Preprocessing* process. The *Complete Login* process uses it for user authentication. The user profile data contains information such as the user's name, email, role, preferences, and history of queries. For example, the *Input Preprocessing* process can use the user's role to determine what kinds of queries they are allowed to run, or use their preferences to customise the user interface.
- **Simulation Data** – holds all data related to the simulations, including the results of the analysis. This data store shows the EDD/AQIS interfaces of the EXA4MIND platform. The data store contains information such as the parameters, inputs, outputs, and metadata of each simulation run. For example, the *Analysis Data Selection* process may use metadata to filter and select relevant data for the current analysis workflow.

2.1.3 Processes and Data Flows

In this section, the individual processes presented in Figure 1 are described, starting on the left side of the diagram.

Complete Login

This process corresponds to a common user authentication used in modern applications.

Input Preprocessing

This process handles user queries and data, and processes them for further use. The *Input Preprocessing* process receives the User Query data from the user and validates it according to the rules and constraints defined by the system. It also receives the User Profile data from the User Details data store and uses it to customise the query processing. For example, it may check if the user has enough privileges to run a certain query, or if the query is well-formed and complete. This process can be further decomposed to several sub-processes as shown in Figure 2.

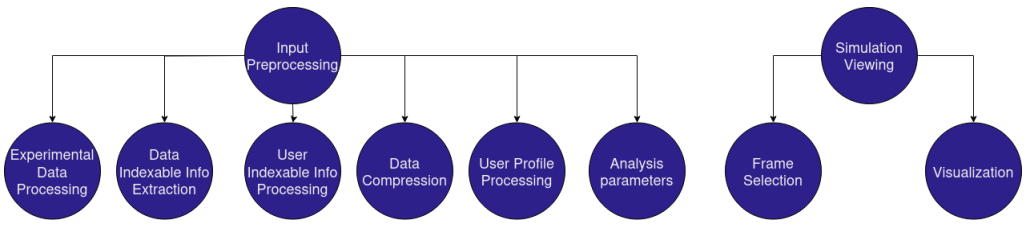

Figure 2: WP4 Input Preprocessing and Simulation Viewing Decomposition

Analysis Handling

This process coordinates the analysis workflow life cycle. It takes in processed user queries, requests the required data from the *Analysis Data Selection* process, passes both into *Script Creation* process and after the analysis is done, saves the results into a data store, and provides the results to the user.

The *Analysis Handling* process acts as a central controller for managing and executing different types of analysis workflows. It also handles errors or exceptions that may occur during the analysis process. For example, it may send a notification to the user if an analysis workflow fails or takes too long to complete.

Analysis Data Selection

This process selects relevant data from the data store for the current analysis and, if needed, selects data to visualise for *Simulation Viewing* process. Relevant data can be either molecular dynamics (MD) simulation datasets, experimental data from the lab or even a file with weights. Figure 3 shows detailed decomposition of this process.

Simulation Viewing

This process launches an application to visualise snapshots from MD simulation to interactively select relevant parameters for further analysis. The GUI can also visualise fluctuations of relevant data, i.e., MD simulation datasets, experimental data, and weights. This process can also be further decomposed to several sub-processes as shown in Figure 2.

Script Creation

Based on the Analysis, Parameters, and Selected Data, this process creates relevant scripts to be executed for the user-defined analysis. At this point, we assume that user either directly specifies parameters which he wants to monitor or chooses available experimental data for the comparison. Refer to Figure 3 for further decomposition of this process.

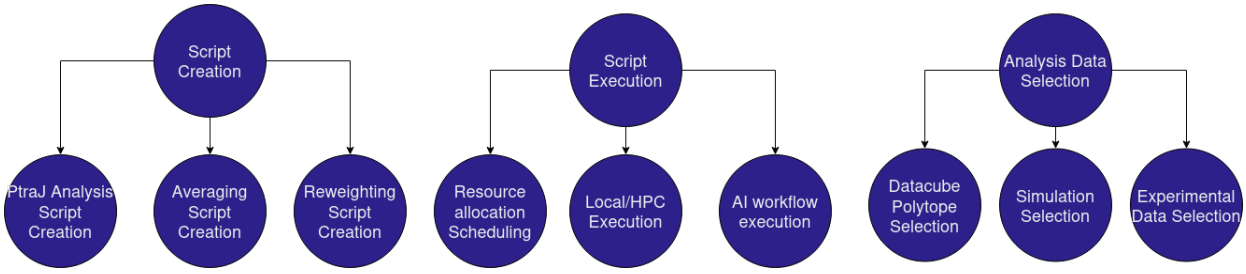

Figure 3: WP4 Script Handling Processes Decomposition

Script Execution

This is the process that executes the script, distributes the workload, and provides the results back to the *Analysis Handling* process. It receives the scripts from the *Script*

Creation process and executes them on the available computing resources. It may use parallel or distributed computing techniques to speed up execution and handle large amounts of data. It then sends back the results of the execution to the *Analysis Handling* process for further processing. Figure 3 shows detailed decomposition of this process.

Data Flows

The Data Flows connecting these processes start with the User generating a User Query that gets processed in the *Input Preprocessing* process. If the query requires uploading new data into the Simulation Data data store, Processed Simulation Data saves it into the database. After preprocessing, the Processed User Query is passed to the *Analysis Handling* process.

Analysis Handling process requests required data through Analysis Data Request from the *Analysis Data Selection* process. If the User requires to see a frame of the simulation, the *Analysis Data Selection* process requests a frame of the relevant simulation and passes it to the *Simulation Viewing* process through the Data To Visualize data flow.

The user specifies his data query and passes them through the Data Selection back to the *Analysis Data Selection* process. The *Analysis Data Selection* process extracts the Selected Query Data for the chosen analysis from the Simulation data and passes it back to the *Analysis Handling* process through the Selected Data data flow. The *Analysis Handling* process then passes the Analysis Parameters and Selected Data to the *Script Creation* process.

The created Analysis Scripts are then passed to the *Script Execution* process, and the Analysis Result is then passed back to the *Analysis Handling* process. The Analysis Result is then saved in the Simulation Data store, and the Query Result is passed back to the User.

2.1.4 Computational Aspects

The application case involves handling a large amount of data from hundreds of simulations, each ranging from GBs to TBs in size. The main computational challenge is to perform many different analyses on these simulations which can be computationally intensive and time-consuming, depending on the complexity and size of the data. Thus, the application case will optimize the analysis workflows and leverage the HPC resources as much as possible. Additionaly, we intend to run new simulations to validate and refine our reweighting algorithms. Running new simulations requires a large amount of computational resources. Therefore, the application case will also require a connection to HPC clusters to speed up the process.

2.1.5 Example Workflows in Dataflows

- **Upload of a new Simulation** - User requests to upload a new Simulation to the Simulation Data store. The User Query contains the Simulation files in a predefined format, including additional information regarding the simulation required for indexing. The *Input Preprocessing* process checks the validity of the

files and if the user is authorized for this upload. It also extracts key information from the data for easy identification of the Simulation. It saves the result into Simulation Data store and reports the status of the upload back to the User through the *Analysis Handling* process.

- **User defined analysis** - User requests to create a new analysis on a specific simulation without the exact knowledge of simulation parameters. The User Query contains a request to view a part of the selected Simulation. The *Input Preprocessing* process checks if the user is authorized for this action and passes the request to *Analysis Handling* process.

  Analysis Handling process passes this information to *Analysis Data Selection* process where a slice of the requested simulation is extracted and passed to *Simulation Viewing* process. The user then explores the slice of the simulation and selects relevant parameters for further analysis and defines the information extraction methods.

  The *Simulation Viewing* process passes these parameters to the *Analysis Data Selection* process. *Analysis Data Selection* process extracts and passes the data to the *Analysis Handling* process, which then passes it to the *Script Creation* process. The *Script Creation* process creates the required scripts for this analysis and passes them to the *Script Execution* process. The *Script Execution* process allocates the resources and runs the analysis locally or on the HPC infrastructure.

  After the analysis is done, the results are passed back to the *Analysis Handling* process, which saves them into a data store and also informs the user about the successful execution.

- **Reweighting** - The user requests to calculate reweighting of a simulation into a new force field. The User Query contains the request of the operation on a source simulation using a target force field, both already existing in the data store. The *Input Preprocessing* process checks if the user is authorized for this action and passes the request to the *Analysis Handling* process. The *Analysis Handling* process requests the data required from the *Analysis Data Selection* process, which extracts the simulation and force field data from the data store and passes it back to the *Analysis Handling* process. The *Script Creation* process creates the required scripts for this analysis and passes them to the *Script Execution* process.

  The *Script Execution* process allocates the resources and runs the analysis locally or on HPC infrastructure. After the analysis is done, the resulting weights are passed back to the process, which saves them into the Simulation Data data store and also informs the user about the successful execution.

2.1.6 Key terms glossary

Additional information about the terms used in the glossary can be found in an Amber manual (Case et al. 2022).

- **Simulation**
  - Molecular dynamics (MD) simulations predict how every atom in a protein or other molecular system will move over time, based on a general model of the physics governing interatomic interactions (Newton's equation of motions).
  - Simulations are run in a specific empirical potential (force field).
  - Simulation outputs Trajectory (consecutive snaphots with xyz coordinates for every atom) and Log (outputs with energies for every force-field term) files.
- **Trajectory** Sequence of snapshots of a simulated molecular system that represents the atomic coordinates at specific time intervals, allowing for the investigation of the dynamic behavior and properties of the system over time.
- **Topology** Description of the connectivity and types of atoms, bonds, angles, dihedrals, and impropers in a molecular system, based on a chosen force field. Contains only partial (incomplete) information about the force field as it is specific for a particular system.
- **Force field** Mathematical description of the potential energy and forces between atoms or coarse-grained particles in a molecular system, based on empirical parameters or quantum mechanical calculations. Force fields define the interactions between particles in molecular simulation and ultimately determine the quality of any simulation. Force field consists of two files: Prep file and a Parameter file.
  - **Prep file** Defines the atom names, types, charges, and connectivity for each residue or molecule in a molecular system (nucleotides and amino acids for nucleic acids and proteins, respectively).
  - **Parameter file** Specifies the values of the force constants, charges, Lennard-Jones parameters, and other parameters for each type of atom, bond, angle, dihedral, and improper in the force field. Chang 2005.
- **Solvent model** Each force field can be used with various solvent models (some of them are recommended for a particular force field). Solvent models differ in parameters.
- **Ion parameters** Each solvent model has recommended ion parameters. Typically, monovalent Na+ or K+ ions are used for neutralization and most of simulations are run at 0.10-0.15 M KCl/NaCl salt (or 1.0 M salt excess).
- **Additional force-field terms** Force field modifications that are orthogonal to existing terms (i.e., general H-bond fix (gHBfix), nonbonded fix (NBfix), ect.). They will be considered at a later stage.
- **Experimental data** Measurements obtained from experiments that probe the structure, dynamics, or function of a molecular system, such as nuclear magnetic resonance, X-ray crystallography, fluorescence spectroscopy, etc.

- **Analysis** The process of extracting useful information from molecular dynamics simulations, such as structural properties, thermodynamic quantities, kinetic rates, conformational changes, etc. Analysis always requires Trajectory file and Topology file. Analysis consists of the following information extraction methods:
  - Distance,
  - Angle,
  - Dihedral,
  - RMSD, RDF, etc. (more functions will be added later),
  - Outputs Analysis output file.
- **Analysis output** Contains the results of an Analysis run on Trajectory and Topology file.
- **Reweighting** Technique to extract weights for molecular configurations sampled by simulations. Reweighting can re-evaluate the results of MD simulations under the assumption of a modified force-field parametrisation without repeating the entire simulation. Experimental data are used for evaluation. It is a process that requires a topology file from Source simulation file and Target force field files, and an Analysis output file. The Analysis that creates the Analysis output file is based on the similarities of the topology of Source simulation and Target force field. Outputs Weights output file.
- **Source simulation** Simulation, in which we are interested to see the results of the analysis made under a Target force field.
- **Target force field** A force field that we want to use to modify the results of an analysis made based on source simulation.
- **Weights** A file obtained by the Reweighting technique, which contains the weights of the molecular configurations sampled by the simulation. Weights are used to improve agreement between simulation and experiment or correct for systematic errors in the force field. The weights file consists of numerical values that indicate the reweighted probability or importance of each configuration; these values can be used with the analysis output of the source simulation.

2.2 Industry Application Case - Valeo (WP5)

The main focus of this application case is scaling a data processing pipeline focused on producing AI models for autonomous driving sytem validation. The expected data volume for this case is up to PBs and includes raw video recordings from vehicles' onboard cameras and LiDAR (Light Detection and Ranging) point clouds from a huge number of locations and environments. These raw data must be indexed appropriately, given the amount of them; even annotation is performed mainly by ML models inference. Indices will then be used to efficiently generate answers for natural language queries that will enable exploration of the data and create selections relevant for smart system testing.

This application case also places great emphasis on the traceability of all steps to ensure the security of the entire solution and quick reaction to identified Advanced Driver Assistance System (ADAS) failure cases, which require retraining the models and an efficient extension of the training data sets.

Initial data flow of WP5 industrial case is present in Figure 4. The boxes represent individual processes, with arrows describing the data flow. Columns here represent a logical divison of the collaboratioin on implementation of the processes between project partners.

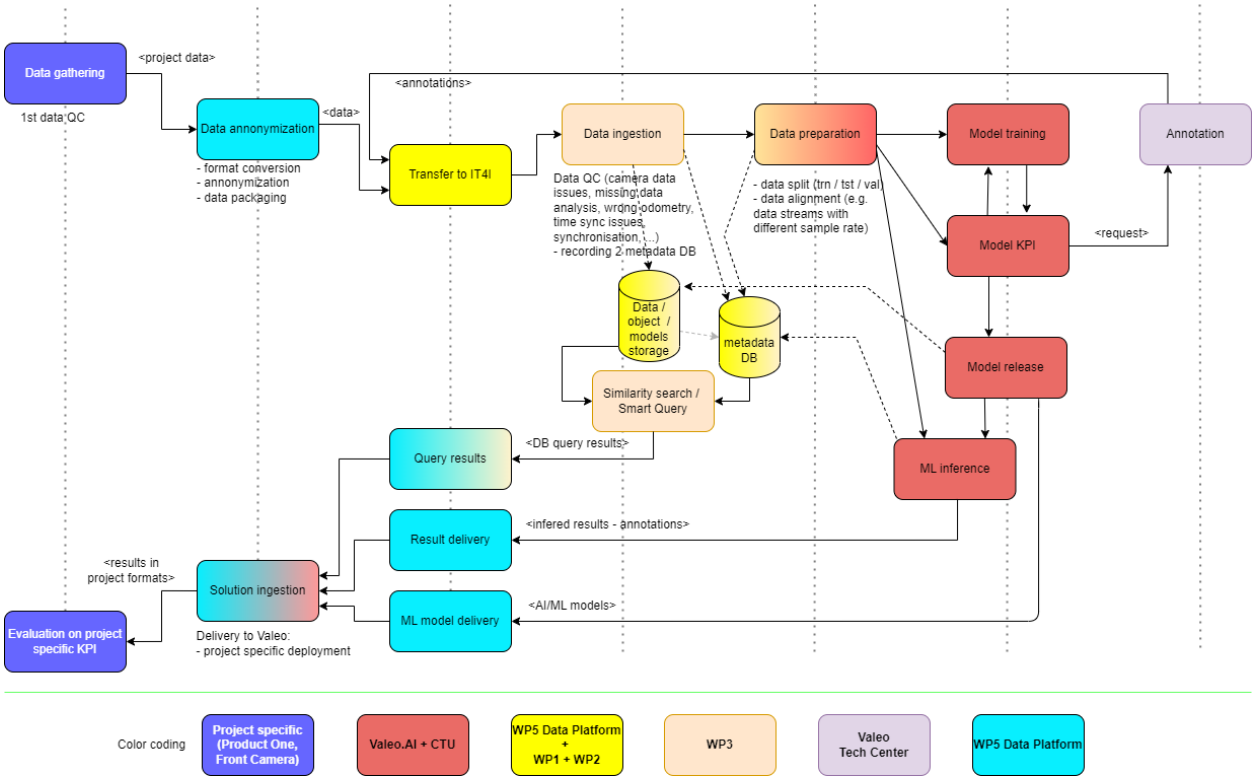

Figure 4: WP5 Data Flow Diagram

2.2.1 Entities

The platform will be used by various users internal to VALEO. For simplicity we will discuss just three user roles. The first one is a **system validation engineer** who will use the platform for preparation of statistical validation campaign (real traffic recordings) planning and monitoring according to required distribution of various objects present, daytime and weather conditions, road conditions etc. He/she will also search for specific classes of objects, maneuvers and situations to explore if the recordings cover various situations and special cases.

The second role is a **VALEO AI researcher** who will use the platform to extract subsets of independent data that either come from the same distribution of meta-data (training, validation and testing sets for ML model training and evaluation) or conversely to find distribution data with respect to some trained ML model to test its

performance in extreme cases (unknown to the model) or to select additional useful training data to boost the ML model performance.

The third type of user is an **annotator** who will be searching for specific situations and captures in order to prioritize his/her annotation tasks given project needs (e.g. now the priority is to annotate captures from tunnels or parking lots). This list is not exhaustive; Valeo will have many more internal user roles.

2.2.2 Data Storages

Data, Object, Model Storage

Here the raw data captured by the reference sensors will be stored. The data will be composed of sets of images from cameras, LiDAR point cloud frames (paired or not), odometry information and, if available, also corresponding annotations. ML model releases along with their training setups will also be stored here.

Metadata Database

Metadata (see glossary in Section 2.2.5) database will be filled with the results of various ML models applied to the raw captures. This will form a rich set of information extracted from the reference sensor data and will be subject to smart querying.

2.2.3 Processes and Data Flows

Data Gathering

This is a project-specific planning process of the data collection campaign. This includes everything from car sensor setup (cameras, LiDARs, etc.), numbers of kilometres, and a specific location where the recordings occur. The recording campaign is ongoing and will continue throughout the life of the project. This will generate the raw data that will be processed within the EXA4MIND project. This is done purely on the VALEO side.

Data Anonymisation

This process performs detection of human faces and vehicle licence plates followed by blurring of respective image areas. This step is necessary to comply with GDPR rules, which then allow the transfer of data to IT4I@VSB infrastructure in the next step.

Transfer to IT4I@VSB

Transfer to storage and compute resources is the act of sending the anonymised data via the VALEO S3 protocol and receiving them on the IT4I@VSB side and delivering them to the desired project storage. There will also occasionally be manual data annotation input, which will be uploaded apart from the raw captures.

Data Ingestion

This process consists of various data quality checks. The raw data together with the results of the quality checks will then be fed via this module into the Data / Object / Model storage creating the base of data, which will then be processed during ML models training and released models inference.

Data Preparation

Data preparation consists of preparing independently and identically distributed sets of training, validation, and testing data. These data splits will be subsets of all uploaded data and will be done repeatedly in different variations. Information about the individual samples present in these sets will be stored in the Metadata Database for every model release. The splits that will lead to incorrectly or unsuccessfully trained models will not have to be stored in the database.

Model Training

Model training takes input training and validation data and performs ML tasks in many iterative steps. This process requires GPU accelerated compute nodes (HPC cluster Karolina). Training and validation data must be temporarily stored in fast storage usually called *scratch*. Data are repeatedly loaded from the disk and used by the continuously trained machine learning model. Model training has a more detailed Data Flow Diagram shown in Figure 5.

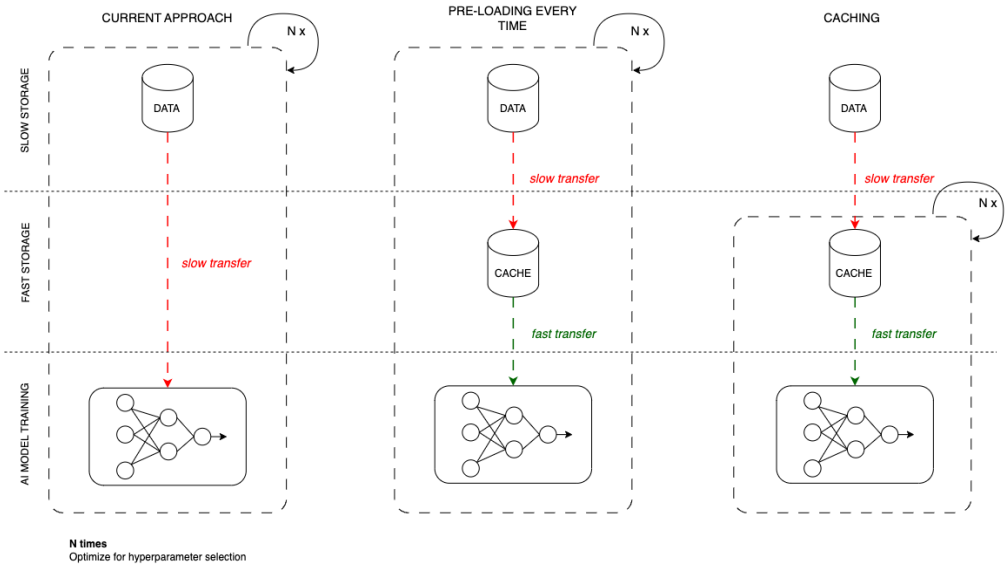

Figure 5: WP5 AI Training

In order to be able to train the models effectively, we need to have fast access to the data, such that their loading does not become a bottleneck. Currently, we keep the data in slow storage. Every time we access the data in this type of storage, there is a possible slowdown.

The first way to tackle this is to preload all the data to the fast storage before running every instance of model training. This way, the data is loaded into the fast storage once at the beginning and then accessed in a fast manner. However, there is still a bottleneck in loading the data from slow to fast storage, which needs to occur for every model training.

Therefore, it is best if the data are transferred only once for all the instances of the training, i.e., the data in the fast storage (cache) are available for multiple training instances, which might be executed even days away from each other. This way, we need to perform the transfer of the huge dataset from the slow storage to the fast storage only once, therefore cutting the transfer times significantly.

Model KPI

Model KPI is a process that takes the testing data set and evaluates the performance of a model at input according to some criteria - usually with manual annotations. In order to evaluate performance of a certain ML model, a request for annotation of a specific testing set needs to be created and a list of files for annotation needs to be sent.

Model Release

Model Release will happen iteratively every time the team reaches a certain model performance based on KPI measurements. The model will be stored in a separate database together with the training setup describing the settings and training / validation / testing data lists. Simultaneously, the model with its settings will be transferred via ML model delivery to VALEO.

ML Inference

ML Inference takes at input a certain release of the model and is performed on all or a subset of the data. The results of inference will be used as input to the metadata database and will be the main source that feeds the metadata database. The results of inference may also be transferred to VALEO via the *Result Delivery* process.

Annotation

Annotation via manual labelling will happen on the VALEO side given a list of files that need to be annotated for a specific task (detection, segmentation, etc.). The resulting manual annotation will then be sent to IT4I@VSB infrastructure through the *Transfer to IT4I* block.

Similarity Search / Smart Query

This process will implement the functionality of natural language user-friendly querying of the Metadata database. By nature, the Metadata database will consist of various datatypes with high dimensionality. Searching through so many values through manual entry would be impractical. Therefore, a series of natural language queries

(multiple sentences specifying the search query) need to be used and translated into a database query. Part of this task is also the creation of the user interface.

Query Results

Query Results will take at input the extracted results from the *Smart Query* search, process them in a format acceptable for VALEO and transfer them to the VALEO infrastructure.

Result Delivery

Result Delivery will take as input the results of a specific ML model inference, transform the results into the format acceptable for Valeo and transfer them to the Valeo infrastructure.

Model Delivery

Model Delivery will take as input the results of a specific ML model release, transform the results into the format acceptable for VALEO and transfer them to the VALEO infrastructure.

Solution Ingestion

Solution Ingestion aggregates the resulting models and data and performs the integration of the ML models and results into the respective VALEO projects and the ground truth extraction pipeline.

VALEO has an internal concept of project referring to a particular car data recording and ML model training campaign, which has its own set of KPIs (different from the EXA4MIND project KPIs). These project-specific KPIs are also being evaluated differently from the ML models KPIs.

Evaluation of these project-specific KPIs will serve as useful feedback to the proper ML model development and should be closely aligned with the ML models KPI developed during the EXA4MIND project.

2.2.4 Computational Aspects

Given the sheer amount of data handled in this application case, it is expected that the most demanding part in terms of computing power will be the processes related to AI models (training, validation, inference). These processes are expected to be launched on an HPC cluster with GPU accelerated compute nodes. The demand for HPC resources is also justified by the fact that the application case is going to train and use ML models to generate additional metadata for their raw data. The final ML model will then be continuously released and enhanced as the pipeline will be scaled up to handle an increasing amount of data beyond the PBs scale.

A metadata database intended for content searching and filtering (or its wrapping REST API) shall provide response times for complex multicriterial queries in maximum single seconds. It shall allow horizontal scaling up to 1 billion of metadata records without negative impact on this response performance.

For data distribution analytics the response times shall not exceed 10 seconds even on 1 billion records filtered/analysed on up to 100 different parameters.

2.2.5 Key Terms Glossary

- **Machine learning model** A machine learning model is a key component of artificial intelligence that uses mathematical algorithms to make predictions or classify data without explicit programming. These models learn from training data by adjusting their internal parameters to identify patterns or relationships within the information. The training process involves minimizing the difference between the model's predictions and actual outcomes (annotations/labels) in the training dataset. Once trained, a machine learning model can generalise its knowledge to make predictions on new, unseen data. Various types of machine learning models exist, including linear regression, decision trees, support vector machines, and neural networks, each suited to different tasks. Machine learning models are applied in diverse fields, including image recognition, natural language processing, recommendation systems, and autonomous vehicles, offering solutions to a wide range of real-world problems.
- **Point cloud** Is a three-dimensional representation of the real world, typically generated by a laser scanner or LiDAR system in the context of autonomous driving and various other applications. It consists of a vast collection of individual data points in space, with each point representing a specific coordinate in three-dimensional space (X, Y, Z) along with additional information such as intensity that gives information about particular surface material reflectivity. Point clouds play a crucial role in tasks like object detection, localisation, and mapping, as they allow autonomous vehicles to identify obstacles, road structures, pedestrians, other vehicles, etc. with precise 3D position.
- **ADAS** stands for Advanced Driver Assistance Systems. Such systems are helping the driver with partial automation of various maneuvers, e.g., automated parking, lane keeping assist all the way towards level 3 and level 4 autonomous driving (level 5 - full self driving - validation is out of timeline of this project).
- **Metadata** In the context of WP5 we use the term Metadata for the following meanings:
  - **Data file envelope attributes** such as file name, creation date, size, hash, compression info, encryption info, etc.
  - **Content metadata** to allow searching, filtering and analysis of distribution of data. Content metadata are related to either a point in time (e.g. single camera/laser frame and its contents recognised by Machine Learning Models or signal value in time like vehicle speed, GPS coordinates, etc.) or an interval (start and end timestamps of events recognised by Machine Learning Models like a car overtaking manoeuver, etc.).
  - **Manual annotations** of image/video and point cloud sequences (present objects, actors, scene features). Annotations infered by trained ML models are called pseudo-annotations or pseudo-labels.

Metadata are typically output of specific data processing, are stored in different file types (e.g. JSON or XML) and are imported to database engines to be indexed to allow quick searching and visualisation - e.g. in Elastic / Kibana software tools.

A special class of Metadata is related to the origin of data - so called *Traceability* - e.g. id and version of the workflow, which produced it. Therefore, it can be used to filter files containing a specific software version of an ADAS algorithm and to compare with other versions to understand the differences.

2.3 SME Application Case - Healthcare (WP6)

The main focus of this application case is using the custom ETL pipeline tool to define and launch complex data analytics and transformations over a large amount of sensitive data from the healthcare domain. Users can define their own ETL pipelines using a graphical UI, while the actual data processing is orchestrated remotely in a traceable and secure manner. The data flow of this application case is presented in Figure 6.

2.3.1 Entities

Users

The users interact with the system by creating pipelines, downloading / uploading new data, and querying logs and output. They pass the User Credentials data to an *Authentication* process to authenticate them. They have varying levels of privileges regarding pipeline creation and execution, execution environments, group management, graph visibility, and so on.

They pass Data to create a dataset by uploading files. They can also download logs and results from previous executions of the processing unit. They create a dataflow pipeline through the front-end interface, request its execution, and fix its priority and its scheduling. This pipeline creation consists of a transformation box selection and configuration, scripting actions with internal or external entities, and input and output selection.

2.3.2 Data Stores

S3 Storage

Stores input and long-term output data. The *Data Processing* process uses it as input for its computation. It stores all the output data generated by the *Data Processing* process.

Logs Database

Stores all data related to the pipeline execution context, metrics, environment, and user actions in a database. The *Data Processing* process logs all execution steps and theirs results to this storage along the execution. Execution logs are retrieved from the Logs storage via the dispatcher.

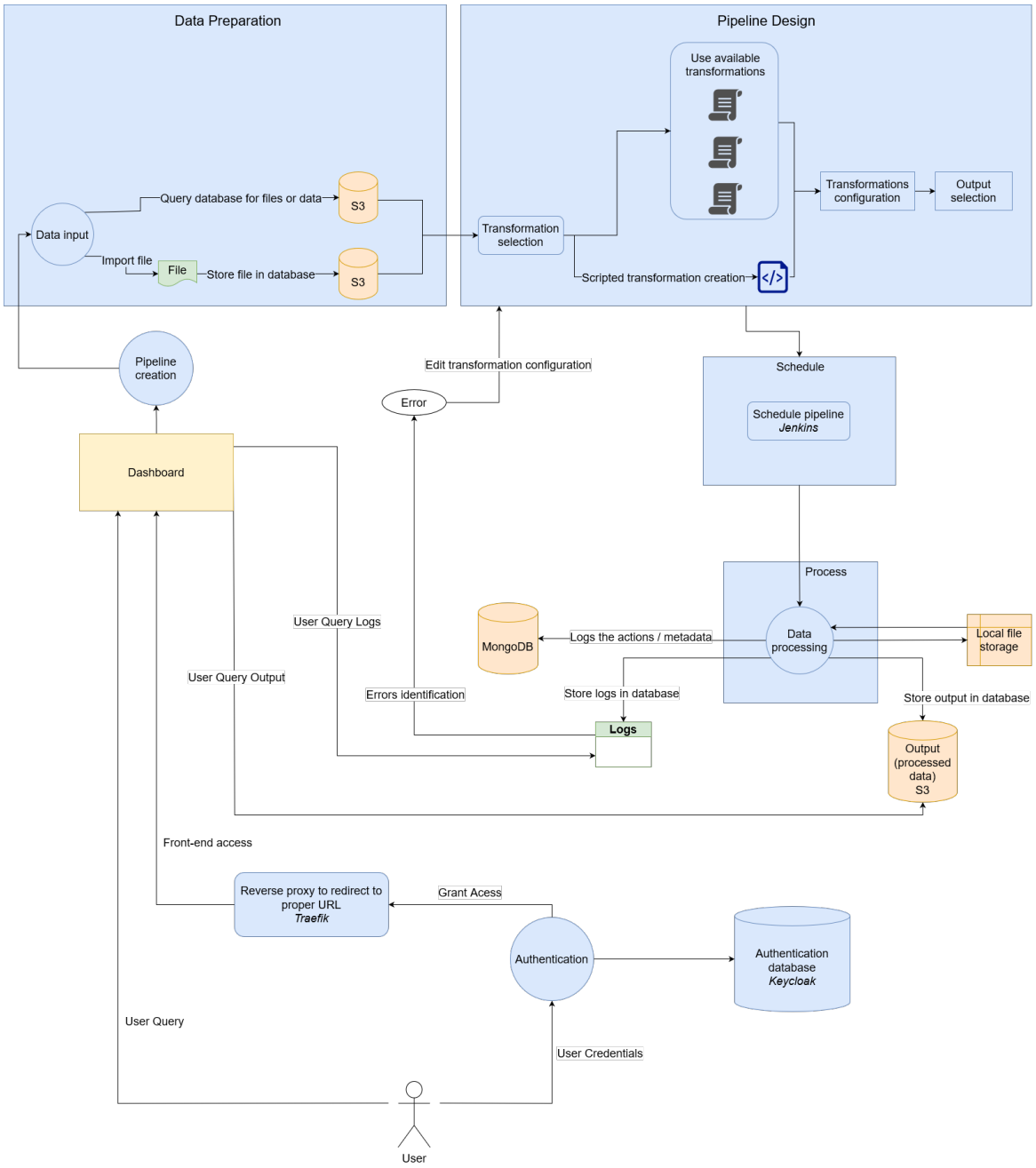

Figure 6: WP6 Healthare Case Data Flow Diagram

Local File Storage

Stores intermediate results of pipeline execution steps. It is used to examine data after execution steps. The local file storage solution is prefered for performance benefits.

Authentication Database

Stores user profile and credentials. The User Profile data contains information such as the user's name, email, role, and group.

2.3.3 Processes and Data Flows

Here we explain processes in the data flow diagram along with their data flow inputs.

Authentication

Process that authenticates Users based on their User Credentials. The *Authentication* process checks if the User Credentials data correspond to the one stored in the Authentication Database. If so, it grants access to the system and redirects the user to the right instance dashboard URL. If not, it denies access and sends an error message to the user.

Pipeline Creation

This process is launched from the dashboard and consists of a graphical design of the dataflow pipeline steps from a starting data source to a final action based on the format of the expected results (file, database, external service, etc.). It manages the user profile and the associated rights (group, environment, visibility). Pipeline creation is subdivided into two sub-processes: *Data Preparation* and *Pipeline Design*.

Data Preparation

This process consists of selecting or uploading files processed by the data flow. The user prepares his data set by uploading the source files to the S3 storage. The user can also use any previously uploaded files (text, images, videos, etc.) in this storage.

Pipeline Design

This process selects and chains transformations to obtain a complete data flow pipeline. The user implements her/his data flow by selecting the transformations box from a catalogue and configuring it with the needed parameters. The user can also script his own transformation box by using a dedicated template box in the catalogue.

Transformation boxes are chained to each other to create a pipeline execution path; all box chains receive output transformation of the previous box in the chain and provide its output to one or multiple following boxes.

The user finishes her/his pipeline design by adding a final step responsible for selecting an output format and destination (file, database, etc.). This step is mandatory in a pipeline design.

At the end of this process, the user passes an Execution Query to the *Dispatching* process to launch the execution of the appropriate pipeline.

Scheduling

This step allows scheduling of pipeline execution and defining their priority. It is possible to set an immediate schedule to start a pipeline execution as soon as possible.

Data Processing

Our architecture allows having several processing units. A new task will always be assigned to the least used unit. In a hypervised system, it would be possible to add processing units on the fly, to adapt to the need.

A pipeline execution produces logs, stored in a database and accessible through our dashboard, and data that may be stored on the S3 file server or on any other system such as an external database or a cloud service, according to the pipeline design.

Our solution differs from many ETLs in the fact that the pipeline is not executed step by step on all the data, but as a continuous pipeline where a piece of output data will be saved before most of the input data is read. The IO throughput being the common bottleneck, this method allows one to optimise the overall processing time. The balance between operations is internally self-managed.

2.3.4 Computational Aspects

The health application case was intended to take advantage of the resources and services offered by the Cloud to handle very large datasets and by the HPC platform for intensive calculation capacities.

In addition to conventional processing of textual data, the application case will implement machine learning models (OCR, CNN) on large quantities of images and PDF files. Storage in the cloud/HPC data centres makes it possible to have a large amount of data that will be used to improve the accuracy of the machine learning models; the training phase is a compute intensive process and requires a large amount of training and test data. The power of the EXA4MIND platform will enable the case to deliver quality results to users within acceptable timescales.

2.3.5 Key Terms Glossary

- **Processing box** A software that transforms input data into formatted output data.
- **Pipeline** Processing box instances linked together to create a path of execution.
- **ETL pipeline** An ETL pipeline is the set of processes used to move data from a source or multiple sources into a database such as a data warehouse.

2.4 SME Application Case - Agriculture (WP6)

TERRAVIEW has a SaaS product called Aquaview. Aquaview enables users to see the relative soil mositure content over an agricultural land. This case works with satellite imagery (geo-spatial) data and custom ML/AI models used to periodically determine various parameters of the soil that can be used to determine its suitability to grow various plants. In each interval, raw satellite data are obtained for selected areas of interest (AoI) and selected timespans from various providers, stored and indexed. Before actual computation of the soil parameters, a set of preprocessing steps is performed, namely cloud masking and water body detection. The results are then stored and indexed appropriately, so that users can access them in a convenient web based UI in the form of map layers.

Figure 7 describes the data flow and transformation steps involved in acquiring data sets from upstream satellite operators, to post-processing resulting in the generation of a relational soil moisture content for a particular geographical area of interest.

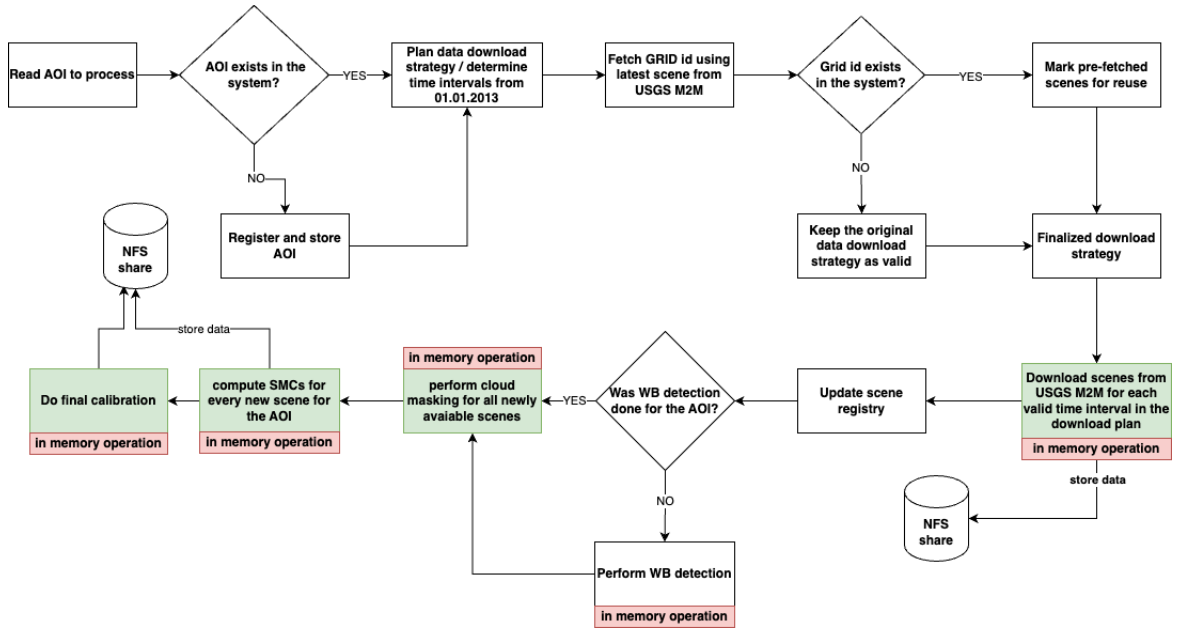

Figure 7: WP6 Data Flow Diagram

2.4.1 Entities

Users

The users are agricultural decision makers, agronomists who use the Aquaview web service to obtain the latest soil moisture content (SMC) analysis for an agricultural plot. Soil moisture analysis helps them understand how to plan irrigation in an optimal manner, as well as aiding in crucial decisions such as when to sow a crop.

2.4.2 Data Stores

High Speed File System

Posix based storage to store raw satellite data, as well as processed SMC analysis results, and metadata files.

PostgreSQL with GIS Extensions

This database is used to maintain links between downloaded satellite scenes and other artefacts necessary for the calculation of SMC.

PostgreSQL with Time Series Extension

PostgreSQL database with time series extension (timescale) is used to store historical weather data signals from our third-party data providers and is continually updated.

2.4.3 Processes and Data Flows

New Area of Interest Registration

Whenever an agricultural land AoI is to be processed and a new SMC is generated, the system checks if the AoI is already registered with the Aquaview platform or not, and depending on the status of the existence check, the satellite data download strategy is formulated. This process has the following steps:

- Verify in PostgreSQL database if the AoI exists in the system or not.
- If not found, register the AoI and store details in PostgreSQL.

Determination of Grid-ID

Grid-ID is a unique reference number assigned by USGS to a geographical area on Earth. For any AoI to be processed, it is important to locate the Grid-ID which the AoI falls under. It could be possible that an AoI is split under multiple Grid-IDs. This process has the following steps:

- Use the AoI to fetch the latest scene available by USGS M2M API – this will help us to find the list of grid IDs from USGS service.
- PostgreSQL DB entry is checked if the Grid-IDs are already known to the Aquaview system.
- If a Grid-ID is already known, the past satellite scenes are likely pre-fetched by Aquaview. These prefetched scenes can be moved into a faster cache for processing a new AoI.

Scene Download Strategy

Since USGS limits the date range in the scene download API, a historical scene download for a new AoI must be performed over several API calls. If a grid-ID is already known in the system then only the delta set of scenes has to be downloaded. This process has the following steps:

- Query the file system / PostgreSQL DB to find if historical scenes for the Grid-ID exist in the system.
- Calculate the delta time window for which the scenes have to be downloaded - starting from 1 January 2013 but from the last available scene (whichever is later) in the Aquaview system to today.
- Perform the scene download from USGS (sequentially due to rate limits enforced by USGS) based on the download strategy – the scenes are stored in the file system, and catalogued in PostgreSQL DB.
- The newly downloaded scenes can be moved to a faster cache to speed up the next processing stages.

Water Body Detection

For our proprietary algorithms to work, we need to locate a body of fresh water close to the agricultural AoI and preferably available in the same scene from the satellite data providers. This process has the following steps:

- Check if the water body (WB) has been located already by querying the DB,
- If a WB exists, no further processing is needed,
- If no WB exists linked to this AoI, then fetch the enclosing satellite scene from NFS into working memory,
- Run the WB detection and validation algorithm,
- Store the detected WB in the PostgreSQL database.

Cloud Masking

Some satellite scenes are rendered unusable due to the presence of clouds and atmospheric disturbances. To improve the user experience where we process only those scenes which have sufficient cloud free data availability, we have to perform cloud masking. This process has the following steps:

- The downloaded scenes are brought from NFS to memory, processed in memory.
- Post processed images are transferred back in file system for the next stage of the processing pipeline.

Soil Moisture Content Tile Generation

The soil moisture content analysis is performed for the desired AoI utilizing the nearby water body and our proprietary algorithms. This process has the following steps:

- The cloud masked tiles are fetched from the network file system,
- The SMC computation is performed,
- The processed and cropped SMC files are stored back in NFS,
- The SMC metadata is stored in PostgreSQL DB.

2.4.4 Post Processing Step

The postprocessing phase prepares the data for easy consumption within the user interface, which includes generation of GeoJSON files to embed the user-relevant data for the SMCs to be shown to the end user.

- Use the SMC tiles as input, generate GeoJSON with metadata, and store the GeoJSON and various zoom level tiles back into the network file system.

2.4.5 Computational Aspects

This case is expected to leverage both on-premises compute resources and HPC. The Cloud resources are necessary to offload periodic download of the AoI raw data from various providers and to launch some light preprocessing tasks. The HPC clusters are then necessary for the actual ML/AI workload, CPU intensive tasks execution such as SMC generation, in parallel and at scale, since the number of AoIs will grow significantly. A significant pool of storage is also anticipated to be used as each AoI amounts to almost 1 TB of satellite data needed for preprocessing and generation of the training data set of the ML workflows.

2.4.6 Key Terms Glossary

- **Area of Interest (AoI)** A closed, polygon region on the planet which is the subject of remote sensing surveys.
- **Soil Moisture Content (SMC)** The relative water content represented in percentage calculated as amount of cubic meters (m^3) of water present in m^3 of soil.

3 Identified Patterns

In this section, we provide analysis of the collected data flows with a focus on identifying a set of common data flow patterns which would be used as an input to the co-design process of the EXA4MIND solution.

3.1 Workflow Execution

All application cases rely on complex, orchestrated workflows across different IT systems providing services or executing computing steps.

This pattern requires a solution for the simple definition and orchestration of inherently complex workflows, including tasks for automatic staging of the data and management of batch jobs on HPC clusters, and tasks for the creation and destruction of cloud entities (pods or virtual machines) for providing services or flexibly-usable computing power.

3.2 Query-to-Computing-System Staging

Almost all application cases express their need for indexing of various kinds of metadata related to their data sets, either to select an input data set for further computational tasks or for analysis of the results.

The pattern identified here expresses the selection of a dataset by a complex database query, whose result can then be automatically staged to a Cloud/HPC compute resource or prepared for download using an appropriate data transfer protocol. The solution implementing this pattern must be able to handle the data volumes required by all application cases. This rationalises the need for the EDD with the AQIS interface.

3.3 DBMS-Dump-to-Computing-System Staging

In special cases, it may be appropriate to transfer the entire database to a target computing system so that the database can later be locally evaluated or queried there.

3.4 Computing-System-to-Database Staging

Results from computations need to be ingested into databases, which is to some extent the inverse process to the *Query-to-Computing-System* staging; however, data ingests are mostly done by insert- or update-like processes (in a RDBMS/SQL-oriented nomenclature), not requiring complex searches over the target databases.

In very rare cases, an entire database might be staged back from computational activities into the EDD.

3.5 Data Ingestion from External Resources

The WP4, WP5, and WP6 Healthcare cases need the data to be available by initiating the transfer from their side (e.g. S3 transfer from VALEO) by an appropriate interface for the nature and volume of the data.

The WP6 TERRAVIEW case expects that the data are being fetched from an external resource by the platform based on the defined set of geographical areas.

In the course of the project, this pattern and its flavour shall be extended through easy read or write access to European data ecosystems, opening up new data sources and possibilities for our use cases.

3.6 Query-to-Open-Data Mechanism

The application cases, and in particular WP4, are committed to publishing result data and bringing them into the European data ecosystem. This requires a mechanism to publish data from the EDD, and to ensure that a persistent copy of the published data can be retrieved – ideally, from the source DBMS itself without creating further copies of the data.

3.7 Additional Identified Features

Apart from the concretely identified patterns, we also identified a set of common characteristics, which do not necessarily correspond to a particular data flow pattern, but provide the functionality needed by the application cases.

3.7.1 Natural Language Queries

The application cases expressed their requirement for the query interface of the EDD, which should work with natural language queries or prompts. For example, the WP5 application case will often require filtering by a particular object which appears on certain roads under certain conditions. Such queries are much easier expressed by natural language queries than complex SQL statements.

Given the latest advances in natural language processing and artificial intelligence, this feature would be of particular interest for the application cases since it can make the access to their data significantly easier. The solution needs to be implemented in collaboration with WP3, which has the development of the Advanced Querying and Indexing System (AQIS) as one of its objectives.

3.7.2 Dataset Caching

When considering DBMS from the HPC perspective, it is important to understand that DBMS are thought of as persistent services (implemented e.g. servers with TCP/IP communication), while HPC systems (“Supercomputers”) classically execute short-lived (hours to a few days), highly parallel applications via queuing systems.

It is technically possible to launch a temporary ’cache’ DBMS server in an HPC job, but there are no mechanisms provided to deploy/fill the DBMS and retrieve data stored at the end of the job. DBMS external to the compute nodes, in turn, often cannot be accessed due to firewall restrictions. As a result, HPC applications typically avoid using DBMS, and DBMS-based analytics applications, ubiquitous in the industry, SMEs and businesses, do not use HPC clusters.

A notable use case of HPC production with databases was implemented at the European Centre for Medium-Range Weather Forecasts (ECMWF). For storing their extensive simulation data, ECMWF uses a set of servers with the distributed database

FDB24 directly connected to their HPC system Smart, Quintino, and Raoult 2019. Additionally, there is broader literature on the benefits and challenges of using different types of databases on HPC systems.

In EXA4MIND, we want to address this topic with a solution allowing Scientists, SMEs, and industry to use DBMS easily with European (and public) Supercomputers. This provides attractive options for Extreme Data analytics.

3.7.3 Incremental Datasets

Similar to the caching mechanism described before, an additional feature must be considered, especially when dealing with ML workloads, which often require subsetting of the data sets for training, validation, and inference subsets. Additional set of operations may be considered as well, to enable for example merging of multiple datasets and similar.

Scaling this task to PBs of data introduces additional obstacles, especially when modifying a data set in the TB range with just a few MBs of additional data to observe how the ML model would be affected.

This property requires a caching feature, which allows performing various set operations on data (intersection, subsetting, etc.) without the need to actually move the same data to a location where they are already present. This feature must be designed together with the caching feature, which would then allow transferring only those parts of large datasets which are not already present in the cache closest to a particular compute resource.

4 Conclusion and Next Steps

In this document, we collected data flow descriptions from all the application cases in the project and presented them in a unified form. In the second half, we provideed the initially identified data flow patterns that should be taken into account when designing the Extreme Data Database within the WP2 scope. We expect that these patterns are not the only ones identified as the design of the platform is being done in iterations by the WP1 co-design effort.

5 References

Case, D. A. et al. (2022). **AMBER 2022 Reference Manual**. URL: https://ambermd.org/doc12/Amber22.pdf (visited on 09/29/2023).

Chang, R. (2005). **Physical chemistry for the biosciences**. University Science Books. ISBN: 1-891389-33-5.

Chinosi, M. and A. Trombetta (2012). "BPMN: An introduction to the standard." In: **Computer Standards & Interfaces** 34.1, pp. 124–134. ISSN: 0920-5489. DOI: 10.1016/j.csi.2011.06.002.

EXA4MIND (2023). **Deliverable D1.1 Application cases and Architecture Requirements**.

LEXIS (2023). **LEXIS Platform documentation**. URL: https://docs.lexis.tech (visited on 09/15/2023).

Smart, S. D., T. Quintino, and B. Raoult (2019). "A High-Performance Distributed Object-Store for Exascale Numerical Weather Prediction and Climate." In: **Proceedings of the Platform for Advanced Scientific Computing Conference**. PASC '19. Zurich, Switzerland: Association for Computing Machinery. ISBN: 9781450367707. DOI: 10.1145/3324989.3325726.

This is the version of the deliverable before the review by European Commission.