Ref. Ares(2025)7317522 - 05/09/2025

Explainable, Safe, Contact-Aware Planning and Control for Heavy Machinery Manipulation and Navigation

D 2.1

Initial Data Management Plan

Project Name	Explainable, Safe, Contact-Aware Planning and Control for Heavy Machinery Manipulation and Navigation
Project acronym	XSCAVE
Grant Agreement no.	101189836
Call	HORIZON-CL4-2024-DIGITAL-EMERGING-01-03
Type of action	HORIZON-RIA
Project starting date	01.12.2024
Project duration	48 months
Deliverable number	D2.1
Deliverable name	Initial Data Management Plan
Lead Beneficiary	University of Tartu
Type	R – Document, report
Dissemination Level	PU – Public
Work Package number	WP 2
Due Date	June 2025
Date	05.09.2025
Version	1

Funded by the European Union

Funded by the European Union under Grant Agreement No. 101189836. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Contributors

- Maria Lutsar (UTARTU)
- Arun Kumar Singh (UTARTU)
- Lili Jiang (UMU)

Reviewers

All partners

Participant organization name	Short name	Country
University of Tartu	UTARTU	Estonia
Tampere University	TAU	Finland
Aalto Korkeakoulusaatio	Aalto	Finland
Toshiba Europe Limited	TEU	United Kingdom
FZI Forschungszentrum Informatik	FZI	Germany
CESKE Vysoke Uceni Technicke v Praze	CTU	Check Republic
KOMATSU Forest AB	KM	Sweden
ALGORYX Simulation	ALRYX	Sweden
Umeå University	UMU	Sweden
Novatron OY	NVTR	Finland
Clevon AS	CLV	Estonia

Abbreviations

AI - artificial intelligence
DMP – Data Management Plan
KPI – key performance indicator
WP – work package
XSCAVE - Explainable, Safe, Contact-Aware Planning and Control for Heavy Machinery Manipulation and Navigation
UTARTU - University of Tartu

Executive Summary

This document serves as the Initial Data Management Plan (DMP) for the XSCAVE project which focuses on developing explainable, safe, and contact-aware AI for heavy machinery in sectors like forestry, earth-moving, and logistics.

The DMP establishes a comprehensive framework for managing the project's data throughout its lifecycle, structured around three core components: the Data Management Plan, a Data Ethics Plan, and an AI & Data Risks Management Plan. The entire strategy is built upon the **FAIR principles** (Findable, Accessible, Interoperable, and Reusable) to maximize the project's impact and ensure responsible data handling.

Key points of the plan include:

- **Data Generation:** The project will generate a diverse range of data, including open-source software, pre-trained deep learning models, simulation environments, and extensive datasets from both real-world and simulated machinery operations (e.g., LiDAR, camera images, robot motion data).
- **Data Storage and Accessibility:** Research data and models will be made findable through institutional and public repositories like Zenodo, where they will be assigned Digital Object Identifiers (DOIs). Project code will be version-controlled and hosted on a central public GitHub organization.
- **Access Policy:** The project adopts a hybrid access model. While the goal is to make most outputs openly accessible under licenses like Creative Commons, some data and results, particularly those from industrial partners, may be restricted to the consortium or released under specific conditions.
- **Quality Assurance:** The plan details robust quality assurance protocols for data, metadata, and code to ensure the reliability, consistency, and reproducibility of the project's outputs.
- **Ethical Framework:** A strong emphasis is placed on ethics. The plan mandates compliance with GDPR, outlines procedures for informed consent and data anonymization, and commits to mitigating bias. Key actions include appointing an **Independent Ethics Advisor** and creating charters for **Transparency & Explainability** and **Data Usage**.

In summary, this DMP provides a detailed blueprint for the systematic collection, storage, protection, and sharing of all data and code generated by the XSCAVE project, ensuring that its outputs are secure, high-quality, and ethically managed.

Table of contents

Chapter 1

1. Introduction

1.1. Overview

This Data Management Plan (DMP) template has been developed for the XSCAVE project to establish standardized protocols for managing project data. It covers the entire data lifecycle from collection and processing to storage, sharing, and long-term preservation. The DMP is designed to ensure that all project data is handled according to best practices, regulatory requirements, and ethical standards.

1.2. Relation to Other Tasks and Deliverables

This DMP template is closely related to other project tasks and deliverables, including:

• Research methodology and data collection protocols

• Ethics and data protection compliance documents

• Risk assessment and management strategies

• Technical infrastructure planning

• Reporting and publication frameworks

1.3. Structure of the Deliverable

This document is structured into three main components:

1. **Data Management Plan**: Covering data lifecycle management according to FAIR principles.

2. **FAIR Data**: Covering the best practices of handling data.

3. **Data Ethics Plan**: Addressing ethical considerations in data handling

Chapter 2

2. Data Management Plan

2.1. Overview

This chapter outlines the core components of the XSCAVE Data Management Plan (DMP) and explains its importance in ensuring effective research data management throughout the project lifecycle.

2.2. What is a Data Management Plan and Why It Is Useful

A Data Management Plan (DMP) is a formal document that describes how research data will be handled during and after a research project. It details the data collection, processing, analysis, preservation, sharing, and reuse strategies to be implemented.

A well-designed DMP offers numerous benefits:

- Improves research efficiency by establishing clear data handling protocols
- Ensures compliance with institutional, funding, and regulatory requirements
- Facilitates data sharing and reuse, maximizing research impact

- Minimizes data loss and enhances data security
- Promotes transparency and reproducibility in research
- Provides a framework for addressing ethical and legal considerations

2.3. Data Findability, Accessibility, Interoperability, and Reusability (FAIR)

The XSCAVE DMP adheres to FAIR principles, which aim to make research data:

FAIR principles
Findable: Data should be easy to discover using persistent identifiers, rich metadata, and registration in searchable resources. **Accessible**: Once found, data should be retrievable using standardized protocols, with clear access conditions and authentication methods where necessary. **Interoperable**: Data should use formal, accessible, shared, and broadly applicable languages for knowledge representation, along with vocabularies that follow FAIR principles. **Reusable**: Data should be richly described with relevant attributes, released with clear usage licenses, have detailed provenance, and meet domain-relevant community standards.

2.4. Data Summary

2.4.1. Purpose of the data collection/generation

The purpose of the XSCAVE project is to push AI approaches, specifically those based on Deep Neural Networks towards real-world deployability in safety-critical applications such as earth-moving, forestry and self-driving vehicle based urban logistics. *The core results of the project take the form of proprietary as well as open-source software libraries for perception, control and physics-based simulation of off-road vehicles, excavators, earth-movers, forestry-forwarder machines etc.* The higher-level AI libraries will be interfaced with either Robot Operating System (ROS) API or other custom developed frameworks developed within the project. Some of the prominent robotic tasks that will be handled within XSACAVE are (non-exhaustive list):

- Development of Physics-based simulators for modeling machine-terrain interaction in challenging environments encountered in forestry, earth-moving and outdoor logistics.
- Deep Neural Networks models which act as the simplified surrogate for physics simulators and predict vehicle (earthmover, forestry, wheeled vehicles) terrain interaction based on 3D sensing using LiDAR/Depth cameras or just monocular RGB camera inputs.
- Reinforcement learning, imitation learning, and model predictive control frameworks for controlling the navigation and manipulation capabilities of different heavy machines used in XSCAVE.
- Use of Large-language and foundational models on video, image, speech and text-data for human-robot interaction and explainability experiments.

Open-source codes in the form of trained Deep Neural Network Models as well as their training pipelines will be the most important form of data that will be generated within the project. The project will also create **extensive datasets** which will be used to train those models for specific tasks within the project, some of which are listed above. Another kind of **generated data** will be software libraries for more conventional types of control and planning algorithms based on model predictive control and trajectory optimization, and system integration.

2.4.2. Types and formats of generated/collected data

In summary, the project will collect/generate the following types of data:

1. Open-source code (SC) : These will be python scripts/notebook files, C++ programmes covering diferent aspects of robot autonomy.
2. Pre-trained DL models (PTM): These are neural network models described in popular frameworks like Pytorch and JAX with trained weights that can be directly used for inferencing. The ONNX or TensorRT version of trained models will also be released.
3. Simulation of assets/environments (SE): A large part of the development work will be first carried out in simulations. In this context, several different terrain conditions will be generated to develop autonomous navigation and manipulation policies for earthmovers, forestry machines, and wheeled mobile robots. Thus, the **core data** in the simulation will be the different terrain environments equipped with different vehicle models.
4. Fully/semi annotated datasets (AD): These will be different sensor logs of earthmovers, excavators, forestry machines and wheeled robots/vehicles operating in simulation or real-world.
5. Robot motion data (MD): These will be logs of position trajectories executed earthmovers, excavators, forestry machines and wheeled robots/vehicles operating in simulation or real-world. Additionally, these will also include the log of actuation input used by the respective machine during operation.
6. Camera images (C): These will be RGB or RGB-D images of the environment that earthmovers, excavators, forestry machines and wheeled-robots/vehicle will capture during operation in simulator or real-world.
7. LiDAR (L): These will be logs of point-cloud data collected by the machines during execution in simulation or real-world.

The types of data that the project will possibly generate/collect are mentioned in more detail in the following Table 1, arranged randomly per-partner. These data/codes will be referred, from now onwards, with the partner acronym followed by the letter preceding each dataset, e.g. TAU.A. It shall be noted here that since the project is still in its first stage of operation that mostly deals with drafting requirements and specifications, the list below is a preliminary one and is highly possible to change. The list will be updated in the next edition of DMP.

Partner	**Type**
TAU	A1. L, C, MD ROS2 bags of large mobile robot terrain traversal (LiDAR, mono and stereo cameras, IMU, GNSS, engine torque readings, steering commands); A2. L, C, MD ROS2 bags of excavator operation (LiDAR, cameras, IMU, GNSS, joystick, pressure sensors, encoder). B. Open-source code C. Pre-trained models
UMU	A. DL models B. Simulation assets/environments C. Fully/semi annotated datasets D. Robot motion data E. Camera images
Aalto	A. Open source code B. DL Models C. Simulation models

TEU	A. Open-source code of parts of the developed system B. Pre-trained DL models for explainability, clarifications, learning from human demonstrations C. Fully/semi annotated datasets of explaining simulated/real robot behaviors and failures containing robot states, images, text, human demonstrations through tele-operation D. User studies/questionaries for evaluating explainability, clarifications, human preference alignment
FZI	A. DL Models B. Simulation environments C. Imitation Learning Demonstrations D. Open Source Code
CTU	A. Open-source code of parts of the developed system B. Pre-trained DL models for world prediction, perception and representation C. Simulation assets/environments for training and testing the DL D. Fully/semi annotated datasets of CTU smaller (proxy) robots and simulated robots D1. Robot motion data collected by onboard sensors and external position reference (trackers, GNSS) D2. Camera images collected by onboard cameras and 3rd person videos D3. LiDAR scans collected by onboard sensors and by external 3D scanners
KM	A. MD/AD driver log (GNSS, Crane position, Engine RPM/tourek, Speed, pressure, Command, etc) B. L/C ROS2 bagfiles (LiDAR, camera, GNSS, IMU)
ALRYX	A. Open-source code of a lightweight, micro, differentiable AGX B. Simulation assets/environments of a ground robot, forwarder, and excavator
NVTR	A. Matlab datasets (gnss, pressure sensor, IMU, encoders) B. ROS2 bagfiles (lidar, camera, gnss, pressure sensor, IMU, encoders)
CLV	A. Model data for the SmartWheel platform B. Experimental data on the motor (for Algoryx for creating sim environment) C. Simulation assets/environments of IndigoTech platform (by Algoryx) D. Documentation and access to Indigo hardware and software E. Vehicle run logs – partly to partners, partly public
UT ARTU	A. DL models B. Simulation assets/environments C. Fully/semi annotated datasets D. Robot motion data E. Camera images/LiDAR point cloud. F. Software for system integration. G. Software for low-level, control, and high-level planning.

Table 1. Data types used in the project per partner.

2.5. Re-use of existing data

XSCAVE will focus heavily on developing novel deep neural network architectures, training which requires a large amount of data. During the initial phase, the data need will be fulfilled by leveraging various publicly available datasets listed in Table 2 and 3. In addition, some 3D models, assets, that are available in either proprietary or open-source software or repositories, will be used. For example, one of the proprietary software will be AGX physics engine that has different terrain and vehicle models. It will be provided by the partner ALRYX to the consortium partners to experiment. The open-source models will be used in accordance with the licensing conditions laid out in the respective repositories.

Task	**Dataset(s)**
Control system development for excavator control	• Existing datasets from previous research projects and other datasets from Novatron R&D will be used for developing excavators low level control system
Bias detection and analysis on training data	• Robotic Operations Performance Dataset • Human-Robot Collaboration Dataset • Industrial Robot Control System Dataset
Outdoor driving	• Various outdoor driving datasets, e.g. Rellis-3D (https://github.com/unmannedlab/RELLIS-3D), ROUGH(https://github.com/ctu-vras/rough-dataset), Monoforce(https://arxiv.org/abs/2309.09007)
Explainability	• RoboFail, REFLECT (https://robot-reflect.github.io/) dataset
Face Detection	• *WIDER Face dataset (http://shuoyang1213.me/WIDERFACE/)* • *Face Detection Dataset and Benchmark (FDDB)(https://github.com/cezs/FDDB)*

Table 2. Datasets used by the functionalities developed within XSCAVE

2.6. Origin and expected size of data, data utility

Partner	**Size**
TAU	A. 1-10 TB. B. Code, estimate not provided. C. Pre-trained models, estimate not provided.
UMU	A - 1 GB DL models B - 10 GB Simulation assets/environments C - 1 TB Annotated datasets D - 1 GB Open-source code
Aalto	A. Open source code 1GB B. DL Models 10GB C. Simulation models 1GB
TEU	A. 10 –50 MB B. 1 – 20 GB C. 20 –100 GB D.10 – 100 MB

FZI	A. Simulation environments - 1GB B. Imitation Learning Demonstrations – 20GB
CTU	A. 10 MB, created by CTU employees with the help of LLMs B. 100 GB, trained by CTU employees from project and public datasets C. 1 GB, created by CTU employees D. 4 TB, recorded by CTU employees, anonymized if needed, ROS 2 Bag
KM	A. 100GB B. 10TB
ALRYX	A. Code, estimate not provided B. Simulation assets and code, estimate not provided
NVTR	A. 1-10GB B. 1-10TB
CLV	A. 10-50MB Model data for the SmartWheel platform B. 1-20GB C. Estimate not provided D. 1MB
UTARTU	A. DL models (less than 1GB per model) B. Simulation assets/environments (10 GB) C. Fully/semi annotated datasets (1TB) D. Robot motion data (1TB) E. Camera images/LiDAR point cloud (1TB). F. Software for system integration (less than 1 GB). G. Software for low-level, control, and high-level planning (less than 1 GB).

Table 3. Expected data size collected or generated by each partner

Chapter 3

3. FAIR data

3.1. Making data findable, including provisions for metadata

The consortium is currently evaluating several options for storage of datasets generated within the project. For example, one option is for each academic partner to use their institutional repositories. For example, the University of Tartu has DataDOI (https://datadoi.ee/) that can be used for storing the datasets generated during the course of the project. Record for metadata generated by UT DataDOI will be available containing following information:

- Identifier (DOI, provided by data center)
- Author (creator) and affiliation
- Data collection title
- Year of publication of the data set
- Publisher (institution, data center)
- Resource Type (data set, image, software)

The consortium is also planning to use universal repositories like Zenodo , Figshare , Open Science Framework which also provides similar metadata types listed above with a searchable DOI

The software and code generated within XSCAVE will be made available through Github (https://github.com/): the world's foremost software development platform. Github provides easy access, and a searchable codebase with proper versioning control. A central XSCAVE organizational Github page will be created that will house all the repositories and codes developed within the project. Furthermore, where applicable, publications will have links to the respective repositories containing trained models, Githubs and datasets.

XSCAVE will establish standardized protocols for code repository organization to ensure data integrity, accessibility, and long-term preservation of digital assets. The organizational framework mandates a hierarchical directory structure with designated locations for source code (src/), documentation (docs/), testing materials (tests/), and configuration files (config/), while implementing language-specific naming conventions that promote systematic file identification and retrieval. Data organization follows modular principles with clear separation of concerns, ensuring that related datasets and code components are logically grouped while maintaining individual file responsibilities that support both current project needs and future data migration or archival processes.

The data management protocol will incorporate comprehensive documentation standards, version control procedures, and security measures to maintain data provenance and protect sensitive information throughout the project lifecycle. All repositories will include standardized metadata files (README.md, LICENSE, CONTRIBUTING.md) that document data sources, usage permissions, and contribution protocols, while version control practices ensure complete audit trails through descriptive commit messages and systematic branching strategies. Security protocols mandate the exclusion of sensitive data from version control systems through appropriate .gitignore configurations and environment variable management, while collaboration frameworks establish clear data access permissions and modification procedures that support both individual and team-based research activities while maintaining data quality and institutional compliance requirements.

3.2. Making data openly accessible

3.2.1. Accessibility of Data and Codes, Repositories and Software

The general approach will be to make codes, trained deep neural-network models, and other software that are the most prominent output, available through the central Github page of XSCAVE. For outputs that are built upon existing codebase, or software modules, licensing provisions of the incorporated code will be followed. With respect to actual datasets, Table 4 provides initial information regarding their open access policy. As can be seen, some aspects of the generated data will have restricted access. For example, this includes the propriety software/algorithms generated within Deliverable D3.1, D6.2.

Partner	**Open Access Policy**
TAU	A. Parts openly accessible, the rest to project partners. B. Parts openly accessible, the rest to project partners. C. Parts openly accessible, the rest to project partners.
UMU	A. DL Models, open access B. Simulation assets/environments, partially open, open to project partners C. Annotated datasets, open access D - Open source code, open access
Aalto	A. Openly accessible
TEU	Parts openly accessible, the rest to project partners.
FZI	Openly Accessible
CTU	A. Parts accessible to public, parts only to project partners B. Parts related to published works accessible to public, the rest to project partners only C. Accessible to the public if the licenses of the resources allow D. Parts accessible to public, the rest to project partners
KM	A. Accessible to project partners B. Accessible to project partners
ALRYX	A. Openly accessible B. Accessible to project partners

NVTR	A. Accessible to project partners B. Accessible to project partners
CLV	A. Accessible to project partners B. Accessible to project partners C. Accessible to project partners D. Accessible to project partners E. Partly accessible to general public, the rest to project partners
UTARTU	Openly Accessible

Table 4. Open access policy per partner

During the initial phase of the project, the dataset and codes could also be made available through the local repositories maintained by each partner. For example, the codes are hosted in the GitHub repository maintained by the partner but publicly accessible. Gradually, these individual repositories will be merged to the central XSCAVE Github page. A dedicated section will be created in XSCAVE website that will provide the list of all the datasets hosted by the partners. *The consortium will also explore the option of universal repositories such as Zenodo for housing the datasets. Zenodo can provide DOI to each dataset, making them easily searchable. The size limit offered by Zenodo (approximately 50 GB per dataset) is most likely to be enough*. Moreover, larger datasets can always be split into multiple parts. In some cases, the dataset could be linked to a particular publication, and the partners concerned can make the data available through their own channels. Even in such cases, they will be linked to the dataset page of the XSCAVE and subsequently also migrated to a central repository such as *Zenodo.*

Each dataset will be accompanied with a documentation detailing how to access-use the dataset. Generally, software, most likely in the form of a Python script, will be part of the documentation. The format of the datasets will closely resemble similar ones already existing in public domain to allow for easy usage.

3.2.2 Data Access Conditions.

The consortium does not foresee the need for a data-access committee as the data collected is not expected to be sensitive in nature. For example, the most important data collected will consist of robot movements in different conditions and the generated sensor and actuation information. Nevertheless, in cases where personal data is collected, for example for AI trustworthiness Experiments (D6.2, D6.3, WP5), GDPR guidelines will be followed for making the data publicly available.

Each partner will lay down the conditions of the data generated during the tasks and WPs. But it is expected that the conditions of access will be very generic and according to the norms of current standards in robotics and AI communities. For example, trained deep neural network models and other perception, planning and control software can be provided under creative common licenses such as CC BY-NC-SA. Alternately, a more custom licensing can be incorporated which provides free usage but only for research and academic purposes with appropriate acknowledging the relevant publications. In such a case, the interested user can sign a licensing agreement and will then be provided with credentials to access the data. Details of this procedure will be part of the data documentation.

3.3. Making data interoperable

The generated datasets and codes will be made available in formats that are commonly used in robotics and AI communities. For example, trained neural network models will be made available in "ONNX" format that will allow its usage in many inference engines such as TensorRT, NCNN, OpenVINO, ONNX Runtime. Similarly, images will be in JPEG/PNG format, numerical data will be released as hd5 or npz files, text data as JSON, 3D assets for physics simulation as XML file . The use of these universal formats will ensure interoperability. The associated documentation will clearly delineate the structure/organization of each dataset; the naming conventions used in the files and codes. Moreover, the software included to load, visualize and manipulate the data will have clear annotation of the shapes and dimensions of the different variables stored in the data (e.g horizon and dimension of the recorded robot trajectories). A short tutorial in the form of video or blog will be created to explain the usage of the dataset and released codes. The Github repository of the released codes will use Markdown language (.md) for updating and maintenance of the documentation.

3.4. Increase Data-Re-usage

According to the grant agreement, the first version of open-source datasets, codes will be released in Deliverable D4.1, D4.4 in M18, while the final versions will be released between M44-M48 through various deliverables. The consortium will make the results available to the public as soon as it is released, and it will be actively maintained (addressing Github issues, updating interfaces) for at least 10 years after the project. As mentioned in the previous subsection, use of universally accepted formats and inclusion of software for reading and visualizing data will ensure increased data re-usage, much beyond the duration of the project.

Each partner will be individually responsible for handling the licenses of the generated datasets, codes, etc. The consensus among the consortium is to use the creative common licenses such as Creative Commons Attribution Share-Alike 4.0 (CC-BY-SA-4.0) or Open Data Commons Attribution License (ODC-BY). However, industrial partners ALRYX, TEU, KM, CLV might have a more restrictive release of some of the project results (e.g see D3.1, D6.2, D8.6).

The quality assurance process followed within XSCAVE will be customized considering the main output from the projects are codes, trained neural network models and datasets of images, robot sensors, and trajectory data. In essence, the following quality assurance steps are envisioned:

Data Quality Assurance

Data quality assurance forms the foundation of reliable deep learning models and encompasses comprehensive validation of data integrity, consistency, and statistical properties. The process begins with completeness validation, ensuring all required fields are populated across the entire dataset while verifying that data collection spans the intended time periods and maintains balanced representation across different classes or categories. Format consistency requires standardizing data types throughout all samples, maintaining uniform file naming conventions, ensuring consistent data structures with identical feature counts and schemas, and validating encoding consistency such as UTF-8 for text data.

Range and boundary validation implements rigorous min/max value checks for numerical features, validates categorical values against predefined vocabularies, identifies outliers using statistical methods like IQR and z-score analysis, and verifies that all data falls within expected domain ranges. Statistical profiling generates comprehensive distribution histograms for numerical features, calculates essential statistics including mean, median, standard deviation, skewness, and kurtosis, identifies and documents class imbalances with mitigation strategies, and analyzes correlation matrices to detect multicollinearity issues.

Quality metrics tracking monitors signal-to-noise ratios for time series and image data, maintains data completeness percentages by feature and sample, tracks duplicate detection rates, and measures annotation agreement scores to ensure inter-annotator reliability. Data preprocessing validation documents and validates all transformation steps, implements inverse transformation checks where feasible, verifies normalization parameters to ensure proper standardization, and checks that data augmentation parameters don't distort key features. Pipeline reproducibility requires version control of all preprocessing scripts, uses fixed random seeds for reproducible augmentations, documents hardware and software dependencies, and implements deterministic data loading orders to ensure consistent results across different environments and executions.

Metadata Quality Assurance

Metadata quality assurance ensures comprehensive documentation that enables reproducible research and informed dataset usage. Essential metadata completeness requires documenting dataset version and creation timestamps, data source and collection methodology, all preprocessing steps with their parameters, train/validation/test split information, class distribution and sample counts, and known limitations and biases. Technical specifications must include detailed data format specifications with schema and data types, hardware and software requirements for processing, expected memory and computational requirements, and compatibility information with major ML frameworks.

Accuracy validation involves cross-referencing metadata statistics with actual data analysis results, verifying that documented preprocessing steps match implemented code, validating that reported performance metrics are reproducible, and ensuring version numbers align across code, data, and documentation components. Consistency checks standardize terminology across all documentation, verify that units of measurement are clearly specified, check that categorical mappings remain consistent throughout, and validate that all references and citations are accurate and complete.

The metadata framework addresses privacy and security considerations by implementing data anonymization procedures, removing personally identifiable information, documenting data retention and deletion policies, and ensuring compliance with privacy regulations like GDPR and CCPA. Ethical considerations include analyzing potential sources of bias in data collection, documenting demographic representation, assessing fairness implications of preprocessing steps, providing guidelines for responsible dataset use, clearly documenting data sources and collection methods, disclosing conflicts of interest or funding sources, and including statements about intended and prohibited uses.

Code Quality Assurance

Code quality assurance establishes robust standards for data processing and model training implementations that ensure reliability, reproducibility, and maintainability. Data processing code structure requires comprehensive validation functions that check data completeness, validate data types, verify preprocessing parameters, and confirm output formats. Error handling implementation includes try-catch blocks for all file I/O operations, input parameter validation with clear error messages, graceful handling of edge cases like empty files and corrupted data, and graceful degradation for non-critical failures.

Testing frameworks encompass unit tests for all data transformation functions, integration tests for complete preprocessing pipelines, property-based testing for data invariants, and performance regression tests for large datasets. Model training code emphasizes reproducibility requirements including setting random seeds for all randomized operations, documenting exact versions of dependencies, implementing deterministic training procedures, and saving and versioning all hyperparameters.

Validation and monitoring capabilities implement early stopping with validation metrics, log training metrics at regular intervals, validate model outputs against expected ranges, and monitor for training instabilities such as gradient explosion and loss divergence. Code documentation standards require clear docstrings for all functions and classes, comprehensive installation and usage instructions, example usage notebooks, complete API reference documentation, and troubleshooting guides. Style and structure guidelines follow established conventions like PEP 8 for Python, implement consistent naming patterns, use type hints where appropriate, and maintain modular, reusable code architecture that facilitates testing and maintenance.

Dataset Release Standards

Dataset release standards establish comprehensive organizational and documentation frameworks that facilitate dataset adoption and ensure long-term usability. File organization follows a standardized directory structure with separate folders for data (train/validation/test splits), metadata (dataset info, preprocessing configs, statistics), code (data loading, preprocessing, validation scripts), documentation (README, data cards, model cards), and licensing information. File naming conventions use consistent, descriptive patterns that include version numbers, avoid special characters and spaces, and employ lowercase underscores for separation.

Documentation requirements encompass dataset documentation through data cards that describe dataset purpose and intended use cases, data collection methodology and timeline, known biases and limitations, ethical considerations, recommended evaluation metrics and baselines, and update and maintenance schedules. Code documentation includes clear installation and usage instructions, comprehensive API reference materials, example usage demonstrations, and troubleshooting guides that address common issues.

Format specifications support multiple file formats when possible, providing both raw data and preprocessed versions, with clear data schemas and framework-specific optimizations like HDF5 for numerical data, Parquet for structured data, TFRecord for TensorFlow workflows, and appropriate formats for different data types (JPEG/PNG for images, JSON Lines for text). Version control and release management implement semantic versioning for datasets with major versions for breaking changes, minor versions for backward-compatible additions, and patch versions for bug fixes, accompanied by detailed changelogs, migration guides, deprecation notices, and clear upgrade paths.

Security and compliance measures include secure data transfer protocols, checksum verification for data integrity, appropriate access controls for sensitive datasets, documented security procedures and incident response plans, and compliance with relevant privacy regulations. The release process coordinates simultaneous release of data, code, and documentation, includes community notification and communication strategies, prepares support channels, and establishes rollback procedures for critical issues.

Quality Control Checklist

The quality control checklist provides comprehensive validation procedures for pre-release verification and post-release monitoring that ensure dataset and code quality meets established standards. Pre-release data validation confirms that all data files load successfully without errors, statistical summaries match documented specifications, no corrupted or unreadable files exist in the dataset, class distributions align with documented values, all preprocessing steps are documented and reproducible, and data splits are properly separated without data leakage between train, validation, and test sets.

Code validation ensures that all code runs without errors in clean environments, unit tests pass with 100% success rates, code follows established style guidelines, all dependencies are properly specified with exact versions, documentation is complete and accurate, and examples and tutorials execute successfully. Metadata validation confirms that all required metadata fields are populated, version numbers are consistent across all components, contact information and attribution are correct, license information is clearly specified, known issues and limitations are documented, and update procedures are clearly defined.

Post-release monitoring establishes continuous quality improvement through regular audits of data quality metrics, periodic validation of reproducibility claims, documentation updates based on user feedback, and performance optimization based on usage patterns. Community feedback integration monitors issue reports and user feedback, tracks usage patterns and common problems, implements version control for dataset updates, and maintains backwards compatibility when possible.

Compliance verification ensures ongoing adherence to privacy and security requirements, regular assessment of ethical implications and bias considerations, maintenance of proper documentation for regulatory compliance, and periodic security audits. Automated quality assurance implements continuous integration with automated testing on data updates, regular validation runs on dataset integrity, automated documentation generation, performance benchmarking on model updates, automated alerts for data quality degradation, and user feedback integration systems that facilitate rapid response to issues and continuous improvement of dataset quality and usability.

3.5. Other Research Outputs

All the research outputs are listed in the previous subsections.

3.6. Allocation of resources

Public datasets will be hosted by each individual partner, through either their organizational platforms or using free repositories like Zenodo. The codes will be hosted on free hosting platforms like Github. Thus, the project does not foresee any additional costs related to hosting the data. The costs of generating, annotating and maintenance of data are already included in the personal cost for each partner.

3.7. Data security

During the initial phase of XSCAVE, when the datasets are hosted by individual partners in their organizational repositories, frequent backups will be created to prevent data loss. Moreover, local storage facilities will be used by each partner to ensure security and recoverability of the data. It is highly likely that Zenodo will be used as the data hostage platform, in which case, it will be possible to leverage the security measures of the said platform. Zenodo is maintained by CERN and thus the data will be maintained as long as CERN exists. Data files and metadata are backed up nightly and replicated into multiple copies in the online system. The backup strategy implements multiple layers of protection to ensure data availability and disaster recovery. Metadata and persistent identifiers in Zenodo are stored in a PostgreSQL instance with a master-slave setup operated on CERN's Database on Demand infrastructure with 24-hourly backup cycle with one backup sent to tape storage once a week. This comprehensive backup architecture ensures both immediate recovery capabilities and long-term archival preservation.

Regarding codes, the chosen option of Github provides highly robust storage that is default backed up at different datacenters around the world. Moreover, Github allows a versioning control that ensures recovery of codes in case of unintentional overwriting. Every partner will follow the best code practices

during local code development and use proper version control before pushing their code to the central repository of XSCAVE.

3.8. Ethics

Ethical considerations are a critical aspect of data management, especially when handling sensitive or personal information. XSCAVE project will proactively address potential ethical and legal challenges related to data sharing, ensuring compliance with all relevant regulations and guidelines.

Compliance with applicable laws and ethical standards will be maintained through regular consultation with legal and ethics experts (including institutional ethics boards) and adherence to institutional policies. The project team will obtain ethics clearance for all research involving the use of real-world sensors (including video, LiDAR, and GPS), which may pose indirect risks of capturing personal data. Key ethical and legal concerns—such as privacy, confidentiality, data protection (e.g., GDPR), bias, and intellectual property rights—will be identified and addressed during the ethics review process.

The project will regulate informed consent appropriately for both retrospective and prospective data. Retrospective data will be shared only in anonymized form, allowing for a waiver of consent where no personal identifiers are present. For prospective data collection, explicit informed consent will be obtained from participants, with consent forms clearly outlining what data is collected, why it is collected, how it is collected, who the data controller is, and whether any joint controllers are involved. Informed consent will be requested whenever necessary throughout the project lifecycle, depending on the activities being conducted. Data acquisition will be limited to information essential to the project's success. All data collection activities will follow strict confidentiality protocols to ensure the privacy of participants. Participants will be informed of their rights, including the right to withdraw consent and the right to request data modification or deletion. Records of consent will be securely maintained to demonstrate full compliance with ethical and legal requirements.

To mitigate bias, the project will employ diversified data sourcing and conduct validation in edge or extreme cases. Model transparency logs and explainability outputs will be integrated into the system design to support interpretability and accountability.

3.8.1. Ethics Guidance During the Proposal

Deliverable 2.2 and corresponding Task 2.1 will formulate the guidelines for the consortium so that the trustworthy AI and the Ethical, Legal, Socio-Economic and Environmental, i.e., ELSE factors are also ensured within the project implementation. To this end, previously funded projects such as the FLEXIGROBOTS, Robotics4EU and AEQUITAS as well as the EU guidelines will be used as a baseline.

Additionally, in Deliverable 1, XSCAVE has already created an independent ethics report that highlights the core ethical issues that could be relevant during project execution. Furthermore, the following guidelines were proposed in the Ethics report

Appoint an Independent Ethics Advisor: This advisor must be in place for the entire project to oversee all ethical issues, produce semi-annual reports, and flag any breaches in real-time.

Create a Transparency & Explainability Charter: Publicly document how the AI's decisions can be audited and include metrics to measure algorithmic bias and fairness at key milestones.

Establish a Data Usage Charter: Detail the types of data collected, how it will be anonymized and secured, and implement clear procedures for informed consent and opt-out options.

Develop an Environmental Sustainability Strategy: Monitor energy consumption, prioritize carbon-efficient infrastructure, and consider the environmental trade-offs of using large-scale AI models.

XSCAVE will follow the above guidelines with regular consultation with the independent ethics advisor.