MIKELANGELO

D7.16

The initial Data Management Plan

Workpackage:	7	Exploitation, Dissemination, Communication and Collaboration	
Partner responsible		XLAB	
Author(s):	Daniel Vladušič		XLAB
	Gregor Berginc		XLAB
	Philipp Wieder		GWDG
Reviewer	Niv Gilboa		BGU
Reviewer	Matej Andrejašič		PIPISTREL
Dissemination Level	Public		

Date	Author	Comments	Version	Status
05. 06. 2015	Daniel Vladušič	Initial draft	V0.0	Draft
19. 06. 2015	Daniel Vladušič, Gregor Berginc	Document ready for review	V0.1	Review
30. 06. 2015	Daniel Vladušič	Document ready for submission	V0.6	Submission

Executive Summary

The purpose of this document is to summarize the Data Management Plan of the MIKELANGELO project. By summarizing, we join all the views on the Data Management plan, which are provided in different paragraphs and articles of Grant Agreement, more specifically Description of Action and provided by the European Commission as overall guidelines. We also provide the initial management plan of datasets used and produced in the MIKELANGELO project.

The document consists of the verbatim information found in the Grant Agreement and the Guideline documents, provided by the European Commission. This information is augmented with the actual data about the MIKELANGELO use-cases and the data the said use-cases will use as input and the data that will be generated. Finally, the document provides explicit information about the data to be shared, ranging from the attributes and location to backup of the MIKELANGELO data.

The Data Management Plan is a living document that will be revised periodically throughout the entire MIKELANGELO project in order to provide information relevant to all interested stakeholders. The plan will also incorporate novel findings and/or specifications of the Open Data Pilot governed by the European Commission.

Table of Contents

1 Introduction 4
2 The Data Management Principles 4
2.1 The Data Management Plan template 4
2.2 The contractual obligations 6
2.3 The DoA data management plans 6
2.3.1 The initial DOA-defined Data Management Strategy 7
3 The MIKELANGELO Data 8
3.1 The data used within the use-cases 9
3.1.1 The Bones use-case 9
3.1.2 The Virtualised Big-Data Software Stack use-case 10
3.1.3 The OpenFOAM Use-Case 10
3.1.4 The Cloud Bursting use-case 10
3.2 The data generated within the use-cases and the project. 10
3.2.1 The Virtualised Big-Data Software Stack use-case 11
3.2.2 The OpenFOAM use-case 11
4 The Data Management Plan 11
4.1 The general approach 11
4.1.1 Scope 12
4.1.2 Assets 12
4.1.3 Risk assessment 12
4.1.4 Definition of controls 13
4.1.5 Validation and approval of controls 13
4.2 Standards, repositories and attributes 14
4.2.1 Standards used 14
4.2.2 Repositories 14
4.2.3 Attributes 14
4.2.4 XLAB’s backup procedure 16
4.3 Summary of results of running an OpenFOAM use case on a simple input case 17
4.4 MIKELANGELO DMP summary table 26
5 Concluding Remarks 30

1 Introduction

This deliverable describes the Data Management Plan of the MIKELANGELO project. MIKELANGELO is voluntarily taking part in the pilot action on open access to research data and will, to fulfill the obligations, provide the data used and generated within the project.

This document is the initial version of the Data Management Plan (as of now, DMP for short), which is to be issued during the lifetime of the project and provide additional data about the project's efforts to appropriately collect, record and share the data.

The main target of the MIKELANGELO's DMP is to ensure the scientific principle of experiment repeatability.

In this initial version, we provide the summary of the DMP principles, taken from the Description of Action (as of now DoA) – topics on the Data Management and Ethics review – and the currently known guidelines, connected with these topics.

After this initial summary, we provide an in-depth description of the data to be used and generated in MIKELANGELO. Finally, we present the Data Management Plan as it is known in its initial version until M6 of the project's execution.

The formal update of this deliverable will be provided at M12.

2 The Data Management Principles

This section comprises the DMP principles, templates, gathered from the guidelines, provided by the European Commission (as of now, EC). It is organised as follows: we first present the DMP template, then contractual obligations (the Grant Agreement article 29.3) and finally, the excerpts from the DoA.

2.1 The Data Management Plan template

The Data Management Plan (DMP) template[1] prescribes the following attributes for each of the datasets used or generated within the project:

- ***Data set reference and name for the data set to be produced:*** Description of the data that will be generated or collected, its origin (in case it is collected), nature and scale and to whom it could be useful, and whether it underpins a scientific publication. Information on the existence (or not) of similar data and the possibilities for integration and reuse.
- ***Standards and metadata:*** Reference to existing suitable standards of the discipline. If

[1] http://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf

these do not exist, an outline on how and what metadata will be created.

- ***Data sharing:*** Description of how data will be shared, including access procedures, embargo periods (if any), outlines of technical mechanisms for dissemination and necessary software and other tools for enabling re-use, and definition of whether access will be widely open or restricted to specific groups. Identification of the repository where data will be stored, if already existing and identified, indicating in particular the type of repository (institutional, standard repository for the discipline, etc.). In case the dataset cannot be shared, the reasons for this should be mentioned (e.g. ethical, rules of personal data, intellectual property, commercial, privacy-related, security-related).
- ***Archiving and preservation (including storage and backup)***: Description of the procedures that will be put in place for long-term preservation of the data. Indication of how long the data should be preserved, what is its approximated end volume, what the associated costs are and how these are planned to be covered.

These attributes should be provided on a case by case basis and should always reflect the current status within the consortium about the data that will be used or produced.

Further data management principles are shown in the list below. We believe gathering of this specific guidelines in one deliverable is important to further guide the evolution of the DMP.

Scientific research data should be easily:

1. **Discoverable**
   DMP question: are the data and associated software produced and/or used in the project discoverable (and readily located), identifiable by means of a standard identification mechanism (e.g. Digital Object Identifier)?
2. **Accessible**
   DMP question: are the data and associated software produced and/or used in the project accessible and in what modalities, scope, licenses (e.g. licencing framework for research and education, embargo periods, commercial exploitation, etc.)?
3. **Assessable and intelligible**
   DMP question: are the data and associated software produced and/or used in the project assessable for and intelligible to third parties in contexts such as scientific scrutiny and peer review (e.g. are the minimal datasets handled together with scientific papers for the purpose of peer review, are datasets provided in a way that judgments can be made about their reliability and the competence of those who created them)?
4. **Usable beyond the original purpose for which it was collected**
   DMP question: are the data and associated software produced and/or used in the project usable by third parties even long time after the collection of the data (e.g. is the data safely stored in certified repositories for long term preservation and curation; is it stored together with the minimum software, metadata and documentation to make it useful; is the data useful for the wider public needs and usable for the likely purposes of non-specialists)?
5. **Interoperable to specific quality standards**
   DMP question: are the data and associated software produced and/or used in the project interoperable allowing data exchange between researchers, institutions, organisations,

countries, etc. (e.g. adhering to standards for data annotation, data exchange, compliant with available software applications, and allowing recombination with different datasets from different origins)?

2.2 The contractual obligations

The contractual obligations are taken from GA - Article 29.3 Open access to research data. This article describes what the Consortium has to do provide its data and how it can protect its results, should such case arise during the execution of the project.

Regarding the digital research data generated in the action ('data'), the beneficiaries must:

(a) deposit in a research data repository and take measures to make it possible for third parties to access, mine, exploit, reproduce and disseminate - free of charge for any user - the following:

(i) the data, including associated metadata, needed to validate the results presented in scientific publications as soon as possible;

(ii) other data, including associated metadata, as specified and within the deadlines laid down in the 'data management plan' (see Annex 1);

(b) provide information - via the repository - about tools and instruments at the disposal of the beneficiaries and necessary for validating the results (and - where possible - provide the tools and instruments themselves).

This does not change the obligation to protect results in Article 27, the confidentiality obligations in Article 36, the security obligations in Article 37 or the obligations to protect personal data in Article 39, all of which still apply.

As an exception, the beneficiaries do not have to ensure open access to specific parts of their research data if the achievement of the action's main objective, as described in Annex 1, would be jeopardised by making those specific parts of the research data openly accessible. In this case, the data management plan must contain the reasons for not giving access.

2.3 The DoA data management plans

In this section we present the requirements on the DMP, as presented in DoA. These requirements are provided mostly verbatim and serve as the baseline, showing the evolution of the DMP itself throughout its updates.

The DoA is to provide 4 versions of the DMP - first in M6 (D7.16 - this deliverable), and from there on, on M12 (D7.17), M24 (D7.18) and finally, at M36 (D7.19). The deliverable reports on data management plan implemented in the project, according to the Guidelines on Data Management in H2020. It lists the types of data and it accessibility.

The Data Management Plan is under Task 7.2, where explicit check of the EC guidelines and appropriate response (the DMP) will be provided. Under Task 7.2 we plan to evolve the DMP

from the initial, simpler version, to a more complete version, evolving the datasets, descriptions and ways of their use.

Within the DoA we provide an important declaration, called ***The MIKELANGELO Open Source Declaration:***

> *The Partners of the MIKELANGELO Consortium commit to release as much of the MIKELANGELO results in terms of software code, as legally possible. The Consortium agrees that the aim of the project is to release the fully functional set of software as publicly available components, where some of them may be closed, due to Partners being subjected to certain legal requirements. The source code of the MIKELANGELO project, not already being part of the established repositories (OSv, KVM), will be put to Github (or equivalent). The scientific papers will be released either under Gold or Green access. The data gathered in the project will be released publicly, under the control of the Data Management Plan.*

Under this declaration, the Consortium intends to comply with the plan and will provide the data and code as much as legally possible - the intent required and followed by the Open Data Pilot and the resulting DMP.

2.3.1 The initial DOA-defined Data Management Strategy

In the following sections, we present the initial Data Management Strategy, as it is defined in the DoA. These sections directly answer the DMP Template, as presented in Section 2.1.

2.3.1.1 Types of data to be generated/collected

The data collected within this project is about the performance and security of various components of the whole MIKELANGELO Cloud & HPC stack, possibly when pursuing most of the efficiency related KPIs that essentially will provide comparisons between different components. The Consortium intends to publish data about the experiments conducted, conditions under which these experiments are done and related benchmarks. In addition to the data itself the Consortium will strive to provide a critical evaluation of the data itself in a short introduction or analysis/conclusion in order to both facilitate the understanding of the data to a person not involved in the data collection as well as simplify the preparation of public deliverables and/or scientific papers.

2.3.1.2 What standards will be used?

As the data collected has no ethics or security issues, we will offer the data in the simplest possible way (possibly even comma separated values - CSV) along with the experiment descriptions. We have decided to start with as simple a plan as possible in order to introduce all partners into the process. As we have already stressed in introductory sections, the plan will be refined and will be updated with a more standard way of publishing data. As for repeatability and stability of the tests, we will store the set-up of the experiment along with the then-used software version. The experiments will be repeated (the usual number of repetitions is 10) in order to provide the required stability of results.

2.3.1.3 Exploitation, availability of data and re-use

The data collected will be used for demonstration and evangelism purposes. It will be made available for independent verification and re-use through the project's website and also through the partners' websites. We explicitly note the possibility that some of the Background data cannot be shared - (e.g. the OpenFOAM use-case will most likely use the data that can be shared, however if the Consortium benefits from the use-case, using the data that cannot be shared, it will pursue that option). Also, there is a possibility that the calculation results of the use-cases cannot be shared, however the performance data will be shared. These specifics for data use, re-use and publishing are provided in detail in the Consortium Agreement, while the general rules on IPR are provided in the Grant Agreement.

2.3.1.4 How will this data be curated and preserved?

Each participant responsible for the experimentation and for measurements and fulfilling the objectives and corresponding KPIs, will adhere to the overall Data Management Plan, describing the ways, policies to gather data in a statistically significant and academically correct way. Preservation will be taken care of as said before - through the project's web site, partners' websites and suggested storage locations.

3 The MIKELANGELO Data

In this section, we provide the in-depth descriptions of the currently known datasets to be used and generated within MIKELANGELO. These descriptions are summarized in Section 4.4, Table 1.

MIKELANGELO will massively use data, especially during the implementation and validation of the four use cases, including cancellous bone simulation, aerodynamic maps for aircraft designs, big data software stack and cloud bursting. Some of the data types to be utilised are currently still not defined as it depends on the implementation of the use case which starts in M12 of the project.

Short overview on the data to be used, generated and re-used:

- Types/Origin of data to be used: Data used is either random or without any possible ethical implications. It is to be used for generating required loads, testing performance and security and further on, for evangelising the results of the project.
- Types of data to be generated/collected: The data about performance and risk assessment of various components. As the data collected has no ethics or security issues, we will offer the data in the simplest possible way (possibly even comma separated values - CSV) along with the experiment descriptions.
- Reuse of data: The data collected will be used for demonstration and evangelism purposes. It will be made available for independent verification and re-use through the project's website and also through the partners' websites.

A more in-depth explanations on the specific datasets used in different use-cases is provided in the following sections – sections 3.1 and 3.2.

3.1 The data used within the use-cases

3.1.1 The Bones use-case

The dataset handed to the project was generated in the following way: The intact femoral head was scanned using a technical micro focus computer tomography system at the "Institut für Bauweisen und Konstruktionsforschung – DLR Stuttgart". The resulting data set is a volume data set consisting of a density field and the header data necessary to describe the regular grid.

Explicitly the source data set consists of 4 files:

1. The density field with 1680 x 1740 x 1752 image points of 4 Byte resolution.
2. The binary header describing the resolution of the regular grid by 3 x 8 byte floating point data.
3. The ASCII header holding the description of the data which is:
4. "Micro-CT of a femoral head taken at the DLR Stuttgart on the 17. February 2010"
5. An ASCII header describing the relevant data chunk positions of the three files mentioned above.

These four files form the so called "puredat" data format which in total consists of approximately 20.5GB of raw data which no longer contain any data related to the patient (neither medical nor personal) from whom the femoral head was taken.

The execution of the use case itself generates continuum mechanical material data of the cancellous bone structures on various resolution scales. These data are purely meaningful for the mechanical analysis carried out with the use case application and are of no medical significance. The data is already publicly available and can be obtained through contacting Ralf Schneider (schneider@USTUTT.de), however we plan to offer it under the MIKELANGELO project facilities too.

The ”Cancellous Bone Simulation” data deals with the development of the material modelling of micro-structured cancellous bone tissues on the continuum mechanical scale from the engineering point of view. For this reason, the existing base data set that is processed within the frame of the MIKELANGELO project,

- is fully anonymised,
- does not include any personal data,
- does not include any data related to patient history,
- does not include any medical data in general.

The specimens used as the basis for the dataset handed to the project were gathered in the following way: During the implantation of a total hip endoprosthesis (an artificial hip joint) the femoral head of the patient gets removed. Normally, the head is destroyed during the removal and afterwards discarded. With a slightly more complex procedure the surgeon was able to assure that the cancellous bone substance within the femoral head was not destroyed during removal and is usable afterwards as the basis for digitalisation.

The procedure was carried out by Prof. Dr.-med. Peter Helwig from the Medical Center – University Freiburg. He also did ask the responsible ethical commission for permission to execute the procedure and took care of requesting the patient's consent. Both, permission by the ethical commission as well as the patient's consent, was given to extract the femoral head by the described procedure and use it further on for research purposes.

The data is currently stored within USTUTT and bound to the same security measures as other data within USTUTT. During the use-case experiments, the data will be used as numerical load in order to generate a comparison data-set for different technology stacks – traditional and MIKELANGELO.

3.1.2 The Virtualised Big-Data Software Stack use-case

The goal of the big data use case is to integrate MIKELANGELO's virtualisation stack with big data technologies. As part of the evaluation of GWDG's big data stack, the stack as a service will be offered to partners within the Max Planck Society, University of Göttingen 19, and the State and University Library Göttingen. Before providing access to the GWDG's system, users will be evaluated based on their data and the goals of their data analysis. This process is executed by GWDG project members according to the data management policy outlined in Section 4.1. GWDG's data protection officer, Dr. Wilfried Grieger, who is also responsible for the University of Göttingen, will supervise this process and assess proposals from users regarding data protection and security issues.

Using the described process, data for this use-case will be gathered. Explicitly, users who will wish to perform analysis on any kind of sensitive/personal data will not be accepted. Since German law requires strict privacy policies when handling personal data, such types of data will be excluded from this use case from the beginning. The Project Execution Board will validate the results and recommendations.

3.1.3 The OpenFOAM Use-Case

The data to be used in this use-case originates from partner Pipistrel. The partner will provide their OpenFOAM calculation data for aerodynamics of the aeroplanes – however, the data in question will be either of no commercial value (e.g., for older aeroplanes; coarser than needed for actual modelling, etc.) or internally protected through IPR agreement in CA.

As the initial case we will study a family of proprietary 2D airfoils (planar contours defined by consecutive points in the plane, from the company library of airfoils) that are used for propeller and wing design of Pipistrel's aircraft. Once the methodology is proven on thissimple case, an example of a complete aircraft geometry (3D CAD model of external surfaces of the aircraft, from the technical database of the company) will be studied. As Pipistrel will be the primary user of this geometry, no intellectual property issues are expected to be encountered.

3.1.4 The Cloud Bursting use-case

The Cloud Bursting use case involves installing Cassandra, and then suddenly putting a huge load on it and seeing how soon the setup can adjust to handle this huge load (i.e., typical cloud bursting scenario). With this use-case, we will not use any sensitive data - the data which Cassandra returns from the queries is immaterial and can be random.

3.2 The data generated within the use-cases and the project.

Use-Cases are managed between the parties providing them and the parties that enable execution of the said use-cases. The data to be generated is performance data of the

MIKELANGELO software stack and data related to risk assessment and security. We will measure automated objectives, e.g., the amount of time it takes to the setup to reach the desired capacity, using the MIKELANGELO software stack - pure performance data of the MIKELANGELO software stack, e.g.: time to create the virtual machine, speed of I/O operations (disk, network), etc. This is the system performance data and will be used for comparisons, evangelism - i.e., publicly available as well.

For all the use-cases, the above described performance data will be publicly available. Given the nature of the input data for the Bones and Cloud Bursting use-cases we can say this data will be shared and the repeatability of the experiments fully supported.

For the OpenFOAM and the Virtualised Big-Data Software Stack use-cases, the generated data may or may not be shared. All cases where data cannot be shared will be appropriately explained and justifications provided.

3.2.1 The Virtualised Big-Data Software Stack use-case

As this use-case will gather input data throughout its execution, a specific selection process has been devised. The selection process will provide screening for any kind of sensitive or personal data that may cause ethical problems for the MIKELANGELO project. However, the actual data to be used in this use-case will target free re-use and sharing, but this cannot be guaranteed. GWDG will pursue the datasets that have the most value for MIKELANGELO. Finally, given the performance data is not critical in any way, it will be treated as such and made available publicly.

3.2.2 The OpenFOAM use-case

The data to be used in this use-case originates from Pipistrel (partner in the Consortium). The performance data (OpenFOAM speed-up, interactivity, faster time to market) will be provided publicly for evangelism purposes. There is a possibility the data to be used within the use-case (aircraft models) cannot be shared further. However, Pipistrel and the whole MIKELANGELO project strives for publicly available data used within use-cases.

4 The Data Management Plan

This section provides the general approach to the data management plan which is then refined into the actual attributes to be collected and shared, repositories, backup procedures and finally, results in summary of the currently known datasets and their attributes within MIKELANGELO project.

This part of the deliverable will be changed throughout the lifetime of the project, reflecting novel guidelines and findings within the project, open access community and even European Commission. A summary of the MIKELANGELO DMP is provided in a form of table in Section 4.4.

4.1 The general approach

A common approach to data management, as it is e.g. implemented by the ISO/IEC 27000 family of standards on information security, includes the following steps:

- Scoping

- Definition of assets
- Assessment of risks
- Definition of controls
- Validation and approval of controls

Information about the different steps is provided in the following sections.

4.1.1 Scope

The MIKELANGELO data management policy applies to all partners of the project "Micro kernel virtualization for high performance cloud and HPC systems" funded under GRANT No. 645402. The data management policy applies for the entire duration of the project.

4.1.2 Assets

The assets covered by this data management policy are the data objects that are used by the use-cases and generated mainly, but not exclusively, as part of the project's Work Package 6 "Evaluation and Validation". Other Work Packages, generating data to a smaller extent, are 2, 3, 4 and 5.

The data used for evaluation and validation is provided by four use cases, as there are:

- Cancellous bone simulation - The Bones use-case
- Calculation of aerodynamic maps for aircrafts - The OpenFOAM use-case
- Cloud bursting - The Cloud Bursting use-case
- The virtualised big data - The Virtualised Big Data Software Stack use-case

The data used and generated is described in Section 3. This kind of data objects is referred to using the term "use case data". The data objects generated within MIKELANGELO project carry information about the performance of the different components that are developed in MIKELANGELO. This kind of data objects is referred to using the term "performance data".

The data management policy covers the assets "use case data" as well as "performance and security related data."

Documents referring to the respective assets are also covered by the data management policy in cases where the same risk level applies to the reference as it applies to the assets itself. Explicitly excluded from the data management policy are any other data or information objects used or generated by the project, including deliverables, software, etc.

4.1.3 Risk assessment

Data management and protection is mainly an issue of risk assessment. As a first step, the data objects have to be classified to define and execute controls. This is best done per use case for the "use case data" and before the actual implementation of the use cases for the "performance data". Through this, the respective controls can be defined for the use cases prior to storing any data objects as part of the work done in Work Package 6. Furthermore, the generation of the "performance data" then follows the data management policy.

A number of different classifications for data categories exist and it is suggested to do a brief evaluation of existing models prior to using one of them in MIKELANGELO. As an example the classification form the Harvard Research Data Security Policy (HRDSP)[2] is provided here:

- Level 1 - De-identified research information about people and other non-confidential research information
- Level 2 - Benign information about individually identifiable people
- Level 3 - Sensitive information about individually identifiable people
- Level 4 - Very sensitive information about individually identifiable people
- Level 5 - Extremely sensitive information about individually identifiable people

In a second step, the respective risks associated with each class of data objects have to be identified. Again, there are already risk analyses done for different levels, also regarding the given example. These analyses serve as input and have to be adjusted to the specific needs of MIKELANGELO, however based on the planned origin and use of data, we only expect the data objects of Level 1 classification.

4.1.4 Definition of controls

Depending on the risk analysis (and on the data security level), controls for storage, protection, retention and destruction of data objects are defined. Such controls have to comply with European regulations (like the Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, taking the current proposal of a comprehensive reform into account) and national law.

In Germany for example, where GWDG, Work Package 6 leader and one of the data centre providers, is located, such regulations are clearly defined in the federal data protection law (*Bundesdatenschutzgesetz*) and the respective framework contracts for data processing include sections on technical and organisational means to protect data, on means for retention or destruction, as well as on the return of data. Same reasoning applies for USTUTT, which is located in Germany and is the second data centre provider. Based on the different regulations and laws, MIKELANGELO will further concisely define the controls to manage the data security classes which apply to the “use case data” and the “performance data”.

4.1.5 Validation and approval of controls

Once defined, the controls have to be validated and approved by the Project Execution Board of MIKELANGELO. Following this step, the controls have to be either changed or can be implemented. All project partners are then involved in the execution of the controls (and therefore of the data management policy). The Project Execution Board monitors compliance with the data management policy and provides input to the evolution of the D7.5, The Data Management plan for MIKELANGELO (i.e., this deliverable).

[2] http://vpr.harvard.edu/pages/harvard-research-data-security-policy

4.2 Standards, repositories and attributes

4.2.1 Standards used

The current standards for data to be used and generated within MIKELANGELO follow the principle of scientific repeatability. This means we're striving for the data and metadata completeness. For the data itself, we're currently targeting the simplest forms of data to be used (as already described - Comma Separated Values - CSV).

For metadata, we shall currently adopt textual description of each of the data attributes (in English language, describing also the minimum, maximum and invalid data marking).

Finally, for experiment data we are targeting the complete experiment description, along with the scripts used. An example of the experiment description is provided in Section 4.3.

Further improvement in this area is to adopt a formalised schema, compatible with the OAI-PMH harvesting mechanisms (either as an add-on, or even, directly, when storing the data itself). This improvement can add to the visibility of the data and the project itself, however with adoption of Zenodo (please see the following section), we are not committing to this improvement and are merely acknowledging the possibility to adopt it.

4.2.2 Repositories

The repositories, guaranteeing the accessibility of the data, as chosen by MIKELANGELO, are:

- Final data and paper storage accessibility: Zenodo - https://zenodo.org/
- Staging data storage: OwnCloud instance at XLAB - owncloud.mikelangelo-project.eu where specific folders[3] are publicly accessible and linked from the MIKELANGELO project website. Data that can be shared publicly, but has not been appropriately validated will be shared here. The data will be clearly marked as not final.

We will offer the data through Zenodo archive and OwnCloud instance. The latter will be used as staging data repository and clearly marked as such. The data submitted to Zenodo will be appropriately referenced as final. Papers, archived at Zenodo, will have data associated with them. Please note, XLAB is backing up intermediate data according to the procedure described in Section 4.2.4.

4.2.3 Attributes

The following is the non-exhaustive list of attributes that are being logged as part of the experiments:

- Input datasets generated and used within the project:

[3] http://owncloud.mikelangelo-project.eu/public.php?service=files&t=b19368219ba31f377561a3113e332956&path=/

    - "random" data for Cloud bursting use-case: since experiments will be repeated prior to being published, the use case provider will have to decide whether the random data is also required for a repeatability of the experiment. Since the MIKELANGELO project is building a general purpose stack, this should not be the case, but will have to be considered during the experimentation period.
    - Bones data and the procedure to access the data are described in more details in Section 3.1.1. In case multiple input data will be used during the validation of the use case, an explicit reference to the data model will be provided.
    - Aerodynamics use-case. Primary input data of this use case is the 2D/3D model of an object being analysed by the use case provider (for example, an airfoil for simple cases or an entire 3D model of an airplane for complex evaluation). Model is also comprised of a series of parameters that specify the execution of the simulation itself (e.g., angle of attack, number of steps to consider, logging options, etc.). These are all part of the data to be provided in compliance with the IPR.
    - Big Data use case will initially focus on random data. Similarly to the Cloud bursting use case, it will be mandatory to provide random data in such a fashion to allow repeatable execution of experiments. In case insensitive real-world data will also be available, the dataset will be provided.
- Generated data
    - Baseline is the starting point for a specific benchmark that we intend to focus on. It must be specific enough to allow validation and repetition thereof. Initial baselines are provided as part of a series of deliverables from Work Package 2 for all different components of the MIKELANGELO stack (hypervisor, guest operating system, cloud & HPC architectures and use cases). Baselines will be updated on a regular basis, taking into consideration also advances in external projects.
    - Benchmark is a specification of the process of measuring and evaluating metrics.
    - Outputs of the tests (needed for validation) are an important artefact of the experimentation procedures. Outputs provide data that may not be interesting to the author of the data package report, but may be observed by a co-worker or even an external stakeholder satisfying the requirement 4 (useful beyond the original purpose for which it was collected) of the scientific research data guidelines. This data will be provided as outputs of Work Packages 2, 3, 4, 5 and 6.
- Set-up of the tests: because MIKELANGELO intends to focus on the provision of performance related data, it is of vital importance to capture data that directly affect the execution time. Examples of such parameters are the following:
    - VM type/creation parameters: KVM is the chosen hypervisor for the MIKELANGELO project. The hypervisor itself may be started using development scripts, Qemu command line API or through a libvirt's XML-based configuration. Exact specification must therefore encapsulate the type of the creation procedure used, command line parameters and optionally (in case they are used) also the accompanying configuration data.
    - Hardware configuration must describe the underlying hardware components, such as CPU, memory and disk. In case of a cluster, it is also important to detail the communication components (Ethernet, high speed Ethernet, Infiniband) and the topology itself.

- Software versions must specify all relevant components such as:
  - KVM version (official version) or KVM revision (when working directly from the source code, a revision from Git must be provided). A reference to the repository should also be made (either official KVM repository or project's internal Git repository).
  - OSv version (official version) or the Git revision number. A reference to the repository should also be made (either official KVM repository or project's internal Git repository).
  - Linux version for baselines and benchmarks focusing on Linux comparison
  - Linux kernel version

Important notes regarding the attributes, availability of the used input data and the collection of the generated data:

- The above set-up follows the basic scientific standards for repeatability of tests. This means, we should store as much data as possible to assist in repeating the test (to a certain degree - we cannot, for example, log the size of L2 cache for all experiments, however if it turns out that such an information is valuable for the experiment itself, it should be used). We also store the commands used to go through the test, preferably as a runnable script also suitable for automation within the continuous integration.
- We store this data for all instances which are published (slides, comparisons, papers, baseline, major developments of code), but not for each test run we're doing internally.

An example of a preliminary document that is still under development is in Section 4.3, showing a summary of experiment and results in running of the OpenFOAM case.

4.2.4 XLAB's backup procedure

The backup procedure for XLAB servers and services, based on the said servers, is sketched in the following bullets:

- The Bacula backup system[4] is used.
- The Bacula servers are located at XLAB's data center, in a protected environment
- Backup is stored on hard-drives
- Backup frequency and type:
  - Incremental backup is performed daily.
  - Differential backup is performed weekly
  - Full backup is performed monthly
- Database backups are made as compressed data dumps and full backups made on daily basis
- Backup versions stored:

[4] http://www.bacula.org/

    - 3 months for low-priority servers
    - Business data stored forever
- Monthly transfer of data to off-site location.

4.3 Summary of results of running an OpenFOAM use case on a simple input case

This section shows the verbatim copy of the experiment log for one of the initial tests using the OpenFOAM case. The log shows currently known relevant details and is to be accompanied by the case data (used and generated) and scripts.

1 Executive Summary

The measurements presented in this document are based on the SysBench tool (https://github.com/akopytov/sysbench). In all tested environments, the tool has been compiled in exactly the same way! For the OSv case, sysbench app has been provided as part of the mike-apps repository (https://gitlab.xlab.si/mikelangelo/mike-apps).

The following table lists only few of the metrics that sysbench tool provides. Further details are given in subsections below.

Sys	CPU	I/O prepare	I/O run
Bare metal	189.3231s	/	1.7034Mb/sec
OSv guest	190.7080s	20.87 MB/sec	653.22Kb/sec
Ubuntu guest	235.9607s	165.63 MB/sec	1.5208Mb/sec

You may also be interested in additional SysBench options which you can see in the manual: http://imysql.com/wp-content/uploads/2014/10/sysbench-manual.pdf

2 Physical host description

All experiments in this document have been carried out on a single host computer running Ubuntu Trusty (14.04.2 LTS). The following are the main HW components:

- CPU: Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz
- Memory: 4x4GiB (DIMM DDR3 Synchronous 1600 MHz (0.6 ns)
- caches
    - L1: 128KiB
    - L2: 1MiB
    - L3: 8MiB

Software components:

- Host OS: Ubuntu Trusty (14.04.2 LTS)
- Linux kernel: 3.13.0-43-generic
- qemu: 2.0.0+dfsg-2ubuntu1.11
- OSv: built from source, exact version within each OSv run in the output, e.g., “OSv v0.20-4-g212f20b”

3 Bare metal running Ubuntu 14.04 desktop

3.1 CPU

```
# sysbench --test=cpu --cpu-max-prime=100000 run
sysbench 0.5:  multi-threaded system evaluation benchmark
Running the test with following options:
Number of threads: 1
Random number generator seed is 0 and will be ignored

Primer numbers limit: 100000
Threads started!

General statistics:
    total time:                          189.3231s
    total number of events:              10000
    total time taken by event execution: 189.3195s
    response time:
         min:                                  18.90ms
         avg:                                  18.93ms
         max:                                  26.66ms
         approx.  95 percentile:               18.97ms

Threads fairness:
    events (avg/stddev):           10000.0000/0.00
    execution time (avg/stddev):   189.3195/0.00
```

3.2 IO

```
# sysbench --test=fileio --file-total-size=150G prepare
```

```
# sysbench --test=fileio --file-total-size=150G --file-test-
mode=rndrw --init-rng=on --max-time=300 --max-requests=0 run
sysbench 0.4.12:  multi-threaded system evaluation benchmark
Running the test with following options:
Number of threads: 1
Initializing random number generator from timer.

Extra file open flags: 0
128 files, 1.1719Gb each
150Gb total file size
Block size 16Kb
Number of random requests for random IO: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
Time limit exceeded, exiting...
Done.

Operations performed:  19623 Read, 13082 Write, 41856 Other = 74561
Total
Read 306.61Mb  Written 204.41Mb  Total transferred 511.02Mb
(1.7034Mb/sec)
  109.01 Requests/sec executed

Test execution summary:
    total time:                          300.0054s
    total number of events:              32705
    total time taken by event execution: 176.8968
```

```
    per-request statistics:
         min:                                  0.00ms
         avg:                                  5.41ms
         max:                                 53.81ms
         approx.  95 percentile:              13.63ms

Threads fairness:
    events (avg/stddev):           32705.0000/0.00
    execution time (avg/stddev):   176.8968/0.00
```

4 OSv guest

Using the mike-apps apps repository, we have built the sysbench image with 200GB.

```
# ./scripts/build image=sysbench fs_size_mb=204800
```

This builds a QCOW2 image, however we wanted to use RAW in order to eliminate parameters that may affect final measurements.

```
# ./scripts/convert raw
```

4.1 CPU

```
lemmy@mike:~/work/osv⟫ time ./scripts/run.py -m 8G -c 1 --nogdb --
novnc -i build/release/osv.raw -e "/usr/bin/sysbench --test=cpu --
cpu-max-prime=100000 run"
OSv v0.20-4-g212f20b
eth0: 192.168.122.15
sysbench 0.5:  multi-threaded system evaluation benchmark


Running the test with following options:
Number of threads: 1
Random number generator seed is 0 and will be ignored


Primer numbers limit: 100000
Threads started!
```

```
General statistics:
    total time:                          190.7080s
    total number of events:              10000
    total time taken by event execution: 190.6985s
    response time:
         min:                                  19.01ms
         avg:                                  19.07ms
         max:                                  26.77ms
         approx.  95 percentile:               19.15ms

Threads fairness:
    events (avg/stddev):           10000.0000/0.00
    execution time (avg/stddev):   190.6985/0.00
```

4.2 I/O

```
# ./scripts/run.py -m 8G -c 1 --nogdb --novnc -i
build/release/osv.raw -e "/usr/bin/sysbench --test=fileio --file-
total-size=150G prepare"
...
161061273600 bytes written in 7360.76 seconds (20.87 MB/sec).

# lemmy@mike:~/work/osv⟫ ./scripts/run.py -m 8G -c 1 --nogdb --
novnc -i build/release/osv.raw -e "/usr/bin/sysbench --test=fileio
--file-total-size=150G --file-test-mode=rndrw --init-rng=on --max-
time=300 --max-requests=0 run"
OSv v0.20-4-g212f20b
eth0: 192.168.122.15
sysbench 0.5:  multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1
```

```
Random number generator seed is 0 and will be ignored


Extra file open flags: 0
128 files, 1.1719Gb each
150Gb total file size
Block size 16Kb
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!


Operations performed:  7349 reads, 4899 writes, 15616 Other = 27864
Total
Read 114.83Mb  Written 76.547Mb  Total transferred 191.38Mb
(653.22Kb/sec)
   40.83 Requests/sec executed
General statistics:
    total time:                          300.0029s
    total number of events:              12248
    total time taken by event execution: 199.4243s
    response time:
         min:                                  0.01ms
         avg:                                 16.28ms
         max:                                438.43ms
         approx.  95 percentile:              26.43ms

Threads fairness:
    events (avg/stddev):           12248.0000/0.00
```

```
    execution time (avg/stddev):   199.4243/0.00
```

Using the --dry-run switch, this is the actual command that has been executed during the I/O sysbench:

```
scripts/imgedit.py setargs
/home/lemmy/work/osv/build/release/osv.raw "/usr/bin/sysbench --
test=fileio --file-total-size=150G --file-test-mode=rndrw --init-
rng=on --max-time=300 --max-requests=0 run"
```

```
qemu-system-x86_64 -m 8G -smp 1 --nographic -device virtio-blk-
pci,id=blk0,bootindex=0,drive=hd0,scsi=off -drive
file=/home/lemmy/work/osv/build/release/osv.raw,if=none,id=hd0,aio=
native -netdev user,id=un0,net=192.168.122.0/24,host=192.168.122.1
-device virtio-net-pci,netdev=un0 -redir tcp:2222::22 -device
virtio-rng-pci -enable-kvm -cpu host,+x2apic -chardev
stdio,mux=on,id=stdio,signal=off -mon
chardev=stdio,mode=readline,default -device isa-
serial,chardev=stdio
```

5 Ubuntu 14.04 Server guest

We have create and installed Ubuntu 14.04.2 Server within a 200 GB image:

```
# qemu-img create ubuntu-1404-server.img 200G
```

```
# qemu-system-x86_64 -hda ubuntu-1404-server.img -boot d -cdrom
./ubuntu-14.04.2-server-amd64.iso -m 8G -vnc :1
```

Using xvnc4viewer we have connected to the setup and left default values - the only specific software that was installed was SSH server.

```
# xvnc4viewer -FullColour :5901
```

After Ubuntu was installed, we have ran VM using the following Qemu command, which is similar to how the OSv was booted. We have used different network settings to be able to ssh into the VM instead of keeping VNC connection open at all times

```
# sudo qemu-system-x86_64 -m 8G -smp 1 --nographic -device virtio-
blk-pci,id=blk0,bootindex=0,drive=hd0,scsi=off -drive file=ubuntu-
1404-server.img,if=none,id=hd0,aio=native -netdev
bridge,id=hn0,br=br0,helper=/usr/lib/qemu-bridge-helper -device
virtio-net-pci,netdev=hn0,id=nic0 -redir tcp:2222::22 -device
virtio-rng-pci -enable-kvm -cpu host,+x2apic -chardev
stdio,mux=on,id=stdio,signal=off -mon
chardev=stdio,mode=readline,default -device isa-
serial,chardev=stdio -vnc :2
```

5.1 CPU

```
# sysbench --test=cpu --cpu-max-prime=100000 run
sysbench 0.5:  multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1
Random number generator seed is 0 and will be ignored

Primer numbers limit: 100000
Threads started!

General statistics:
    total time:                          235.9607s
    total number of events:              10000
    total time taken by event execution: 235.9545s
    response time:
         min:                                  23.20ms
         avg:                                  23.60ms
         max:                                  46.05ms
         approx.  95 percentile:               23.95ms

Threads fairness:
    events (avg/stddev):           10000.0000/0.00
    execution time (avg/stddev):   235.9545/0.00
```

5.2 I/O

```
# sysbench --test=fileio --file-total-size=150G prepare
...
161061273600 bytes written in 927.36 seconds (165.63 MB/sec).
# sysbench --test=fileio --file-total-size=150G --file-test-
mode=rndrw --init-rng=on --max-time=300 --max-requests=0 run
```

```
sysbench 0.5:  multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1
Random number generator seed is 0 and will be ignored



Extra file open flags: 0
128 files, 1.1719Gb each
150Gb total file size
Block size 16Kb
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!

Operations performed:  17520 reads, 11680 writes, 37261 Other =
66461 Total
Read 273.75Mb  Written 182.5Mb  Total transferred 456.25Mb
(1.5208Mb/sec)
   97.33 Requests/sec executed

General statistics:
    total time:                          300.0096s
    total number of events:              29200
    total time taken by event execution: 181.4577s
    response time:
         min:                                  0.00ms
```

```
         avg:                                    6.21ms
         max:                                   68.11ms
         approx.  95 percentile:                14.61ms

Threads fairness:
    events (avg/stddev):           29200.0000/0.00
    execution time (avg/stddev):   181.4577/0.00
```

4.4 MIKELANGELO DMP summary table

The MIKELANGELO datasets and the relevant DMP attributes are summarized in the following table (Table 1). This table presents the outcomes of this document in a compact way and presents the currently known facts about the data within MIKELANGELO project according to the Data Management Template.

Table 1: The MIKELANGELO dataset summary table

Data set reference	Data set name	Data set desc.	Standards & metadata	Data sharing	Archiving and preservation
MIKELANGELO-UC-CLOUD-BURSTING	MIKELANGELO Cloud Bursting use-case input data	The data used for the Cloud bursting case. Data described in 3.1.4.	Data stored as CSV	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org.
MIKELANGELO-UC-CLOUD-BURSTING-PERFORMANCE	MIKELANGELO Cloud Bursting use-case performance data	The use-case performance data.	Data stored as CSV	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org.
MIKELANGELO-UC-CLOUD-BURSTING-EXPERIMENT	MIKELANGELO Cloud Bursting use-case experiment data	The experiment data, as defined in Section 4.3	Data stored as text (description of experiment) and scripts. Description provided in Section 4.3.	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org.
MIKELANGELO-UC-BONES	MIKELANGELO Bones use-case data	The Bones dataset, available from USTUTT. The data is described in 3.1.1.	The data format is described in 3.1.1.	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org. Data stored at USTUTT.
MIKELANGELO-UC-BONES-PERFORMANCE	MIKELANGELO Bones use-case performance data	The use-case performance data.	Data stored as CSV	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org.
MIKELANGELO-UC-BONES-OUTPUT	MIKELANGELO Bones use-case output data	Anisotropic material distributions for malicious bones	Data stored as binary or simple ASCII output	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org. Data stored at USTUTT.
MIKELANGELO-UC-	MIKELANGELO Bones	The experiment data, as	Data stored as text	Open Access, under the	Data to be stored at XLAB's OwnCloud

BONES-EXPERIMENT	use-case experiment data	defined in Section 4.3	(description of experiment) and scripts. Description provided in Section 4.3. The FMPS solver and C as well as Fortran code for Bones UC is excluded and cannot be provided.	MIKELANGELO Open Source policy.	instance (when in staging and later on, when final). Data to be stored at Zenodo.org. Data stored at USTUTT.
MIKELANGELO-UC-OPENFOAM	MIKELANGELO OpenFOAM use-case data	The data is available as the OpenFOAM case. The data is described in 3.1.3.	Data stored as the OpenFOAM model.	Some of the data will be publicy available under MIKELANGELO Open Source policy.	The publicly available data from PIPISTREL: - Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). - Data to be stored at Zenodo.org.
MIKELANGELO-UC-OPENFOAM-PERFORMANCE	MIKELANGELO OpenFOAM use-case performance data	The use-case performance data.	Data stored as CSV	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org.
MIKELANGELO-UC-OPENFOAM-OUTPUT	MIKELANGELO OpenFOAM use-case output data	Numerical output as a result of OpenFOAM simulation.	Numerical output.	Some of the data will be publicy available under MIKELANGELO Open Source policy.	The publicly available data from PIPISTREL: - Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). - Data to be stored at Zenodo.org.
MIKELANGELO-UC-OPENFOAM-EXPERIMENT	MIKELANGELO OpenFOAM use-case experiment data	The experiment data, as defined in Section 4.3	Data stored as text (description of experiment) and scripts. Description provided in Section 4.3.	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org.
MIKELANGELO-UC-BIG-DATA	MIKELANGELO Big Data use-case data	The data will be selected through GWDG's partners - - The data processed as	Data stored as CSV	Some of the data will be publicy available under MIKELANGELO Open Source policy.	The publicly available data from GWDG: - Data to be stored at XLAB's OwnCloud instance (when in staging

		part of the use case implementation.			and later on, when final). - Data to be stored at Zenodo.org.
MIKELANGELO-UC-BIG-DATA-PERFORMANCE	MIKELANGELO Big Data use-case performance data	The use-case performance data - Measurements of the performance of the use case execution. Including low-level monitoring and use-case specific metrics.	Data stored as CSV	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org.
MIKELANGELO-UC-BIG-DATA-OUTPUT	MIKELANGELO Big Data use-case output data.	Results and modified data generated through the use case implementation.	Data stored as CSV	Some of the data will be publicy available under MIKELANGELO Open Source policy.	The publicly available data from GWDG: - Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). - Data to be stored at Zenodo.org.
MIKELANGELO-UC-BIG-DATA-EXPERIMENT	MIKELANGELO Big Data use-case experiment data	The experiment data, as defined in Section 4.3 - Documentation, scripts, and configuration to reproduce the use case experiments.	Data stored as text (description of experiment) and scripts. Description provided in Section 4.3.	Open Access, under the MIKELANGELO Open Source policy.	Data to be stored at XLAB's OwnCloud instance (when in staging and later on, when final). Data to be stored at Zenodo.org.

5 Concluding Remarks

We provide the first DMP, based on the actual research of the available facts within the project (data, generated data, IPR and CA), according to the EC Guidelines and finally, based on the research of the available Open Access Repositories.

We have currently opted to gather all these facts under this deliverable, to aim for completeness of the data within the project and to further refine the possible formats, either internally, to improve understanding (i.e., decrease ambiguity) and also externally, to cater for mining of data (the inclusion into the Zenodo repository).

A summary of the MIKELANGELO DMP is provided in Section 4.4.

MIKELANGELO targets the scientific principle of experiment repeatability and will, based on all the gathered facts, improve and evolve the DMP and the data descriptions, gathering and generation procedures.