Industrial Machine Learning for Enterprises

Deliverable D4.1

Requirements for the IML4E Online Experimentation and Education Platform

Project title:	IML4E
Project number:	20219
Call identifier:	ITEA AI 2020
Challenge:	Safety & Security

Work package:	WP4
Deliverable number:	D4.1
Nature of deliverable:	Report
Dissemination level:	PU
Internal version number:	1.0
Contractual delivery date:	2022-05-31
Actual delivery date:	2022-06-30
Responsible partner:	SILO AI

Contributors

Editor(s)	Harry Souris (Silo AI), harry.souris@silo.ai
Contributor(s)	Janis Lapins (Spicetech), Mikko Raatikainen (University of Helsinki), Dorian Knoblauch, Jürgen Großmann (Fraunhofer Fokus), Donny Thomas Daniel (Siemens), Johan Himberg (Reaktor)
Quality assuror(s)	Ihasalo Heikki (Granlund), Dorian Knoblauch (Fraunhofer Fokus)

Version history

Version	Date	Description
1.0	21-05-22	Version for publication

Abstract

The document focuses on the experimentation platform and training material needed for the purposes of the IML4E project. First, the requirements for materializing the experimentation and training platform are described and an architecture addressing those requirements is presented. Second, a Minimum Viable Product (MVP) of the experimentation and training platform is defined. This MVP can cover the needs of the IML4E partners and at the same time it can be the basis for an experimentation platform that can support enterprises. Finally, in this document a suggestion of technology options and tools that can materialize the experimentation platform is introduced and the future steps of WP4 are presented.

Keywords

MLOps architecture, Experimentation platform, Training material, Pipeline orchestration, Model versioning, MLOps platforms

Executive Summary

In this document, IML4E partners present the experimentation and training platform that is expected to be created during the IML4E project. For that purpose, the partners first describe the requirements that need to be covered for the different Machine Learning use cases, and then the partners describe a high-level architecture of the platform that addresses those requirements. Finally, in the document the technologies that will be used for operating the experimentation and training platform are presented together with a Minimum Viable Product (MVP) deliverable of the platform. In the next steps, IML4E partners aim to materialize the MVP deliverable of the platform and utilize it for the Machine Learning use cases in the IML4E project.

Table of contents

1 Introduction

1.1 Role of this Document

The purpose of this document is to present the basic usage scenarios and requirements that the experimentation and education platform should address for the purposes of the IML4E project. In addition to this, architecture considerations and possible technology choices in terms of tools and a deployment environment are discussed.

1.2 Intended Audience

The intended audience of the present document is composed primarily of the IML4E consortium for the purpose of capturing the baseline of the project that the project will advance. However, this document is public and can provide an overview of the current practices to a reader. This document describes technologies for the technically oriented audience rather than the general public or layman.

1.3 Definitions and Interpretations

Experimentation platform: for the purpose of the project "Experimentation platform" is defined as the environment that partners of the project will onboard data, develop and operate machine learning pipelines with an aim to validate their approaches and methodologies around the end-to-end lifecycle of machine learning pipelines in an enterprise context.

Training / Education platform: for the purpose of the project "Training platform" or "Education platform" is defined as the educational portal where documentation and education material generated during the course of the project is saved and maintained. Purpose of the education material is to help practitioners understand the requirements for developing, operating and maintaining ML/AI solutions in an enterprise context and help them materialize the approaches and methodologies defined in the IMLF4E project.

1.4 Applicable Documents

Reference	Referred document
[FPP]	IML4E – Full Project Proposal 20219
[PCA]	IML4E Project Consortium Agreement

Table 1 - Contractual documents

2 Basic Concepts in MLops

MLOps is a derivate from DevOps, the state-of-the-practice way of working for the technical life cycle to develop and deploy software systems continuously in iterative development. While DevOps initially emerged for software systems and services to streamline the development and operations, MLOps extends DevOps by phases related to, on the one hand, data management for the ML models of the system and, on the other hand, ML model management of the system based on data. This data-centered approach, which is associated with ML introduces new dynamics to the software development process and changes how we engineer ML-enabled software systems (Ozkaya 2020).

A typical ML workflow begins with collecting model requirements where the problem being solved is matched with a suitable model type(s). Once the system requirements are understood, relevant data is collected and cleaned in preparation for feature engineering. In certain problem domains e.g., computer vision, the data processing stage may involve additional steps such as labeling. Feature engineering involves extracting and selecting informative features from the data which are used to train models. Model training is an iterative process that involves re-training the model while tuning the model's parameters to achieve a given model's performance. Suitably trained models are then deployed and monitored for errors when in production (Amershi S. et al). Model errors in production can occur as a result of model drift or concept drift, these are caused by changes in the distribution of training or inference data respectively.

As depicted in the figure below, these Data Management, ML modeling, software development, and System Operations stages of the ML workflow are non-linear. They involve cyclic, iterative steps, including feedforward and feedback interactions between the phases, incremental development to improve the system, and continuous operations (Lwakatare et al. 2020).

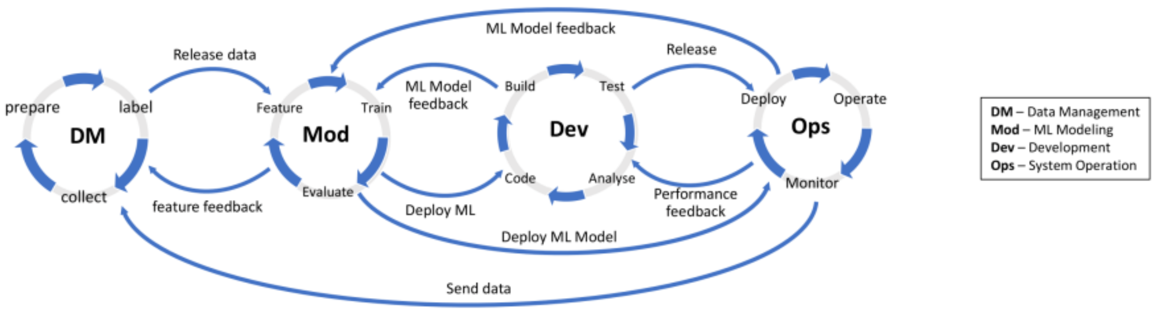

Figure 1: Schematic picture of MLOps (source: Lwakatare et al. 2020).

Although DevOps has matured within the last decade to relatively stable, broadly applied best practices, MLOps is still in its early stages and therefore MLOps best practices are yet to be standardised. Evidence from non-bigtech companies indicates that MLOps practices and tooling tend to vary across organizations (Muiruri D. et.al 2021). Data management related practices can vary as a result of differences in data types, data sizes and annotation formats across neural network architectures. Model training and evaluation commonly require experiment management practices but these are implemented differently depending on the ML domain or team experience with existing tools. Model deployment and monitoring practices are influenced by deployment architectures, whether the model is deployed in the cloud, edge or embedded in other software systems (Muiruri D. et.al 2021).

The figure below refines the above figure with different activities and tools that MLOps can cover by focusing on the data and ML model concerns. The specific activities and tools, as well as related challenges, are covered in more detail in WP2 and WP3 in terms of data and ML models, respectively.

Another peculiarity of MLOps is the inherent experimental nature of activities related to ML. Therefore, the desired level of automation differs so that sometimes fully automated activities and pipelines are not even expected. This is unlike in DevOps, in which full automation is typically preferred.

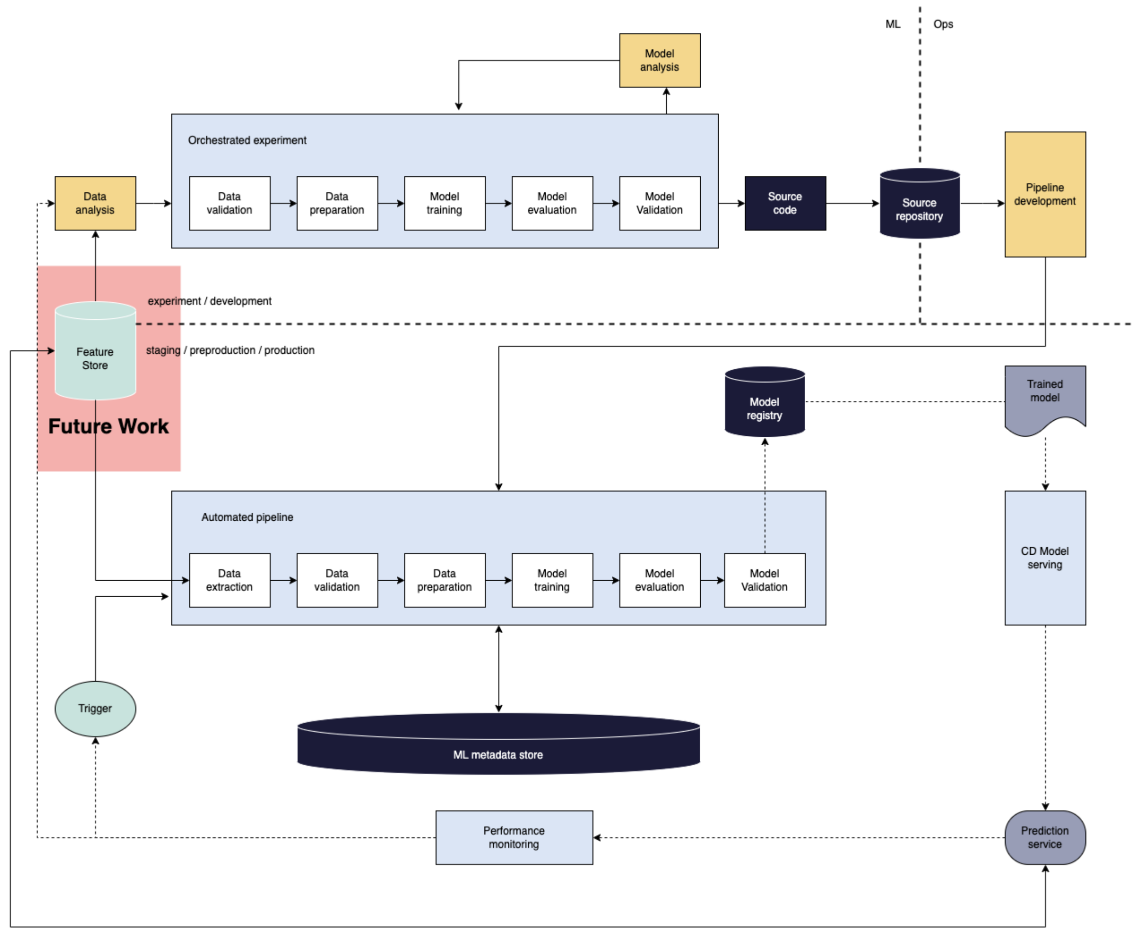

Figure 2: The reference MLOps model in the IML4E project including ML specific activities and tools (Google 2022a) (Licenced under CC BY 4.0 https://creativecommons.org/licenses/by/4.0/).

3 Experimentation Platform Requirements in ML R&D And AI Utilization

In the context of IML4E, the experimentation platform will be in position to provide an environment covering the needs of the end-to-end life cycle of machine learning pipelines/solutions - from their development phase till their continuous execution and maintenance phase. For that purpose, the partners of the project have identified basic usage scenarios and the corresponding requirements that those scenarios bring to the experimentation platform.

It is important to note even at this point that the experimentation platform in this project can be separated into two basic layers:

1. The MLOps tools and common software assets that simplify the implementation and execution of the use cases. Those MLOps tools and software assets help on addressing common MLOps requirements such as automation, versioning and monitoring. A typical MLOps tool that is introduced in the MLOps toolchain of companies is the model registry that enables a unified approach on versioning models and artifacts related to the models. MLFlow is such a tool.
2. The infrastructure, technical layer, that consists of the hardware and network resources for deploying our MLOps tools and for executing the use cases. Example of those resources can be GPU and CPU clusters, data storages, network configurations for controlling access and others.

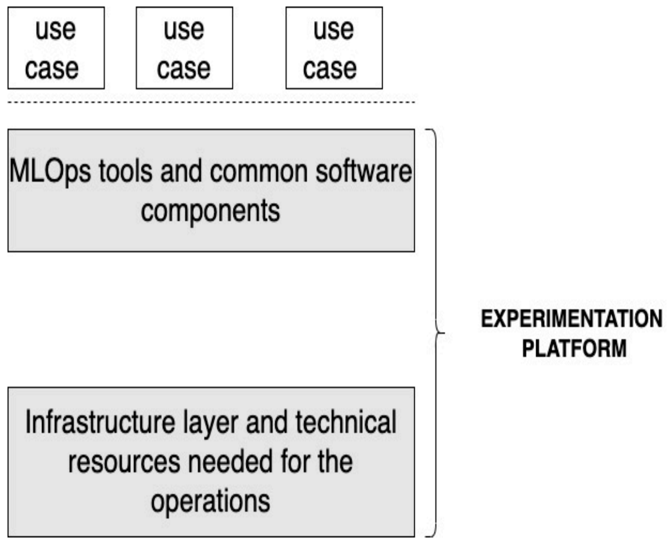

Figure 3: Basic layers of the experimentation platform, on top of those ML use cases are executed

On top of those two layers the machine learning pipelines materializing our use cases will be deployed and executed.

3.1 Usage scenarios and requirements for the experimentation platform

From a high-level view, the usage scenarios of the experimentation platform evolve around the following activities:

1. Data onboarding and management
2. Machine learning development and engineering
3. Machine learning deployment, operation and maintenance
4. Accessing and provisioning resources and infrastructure for the operations of the platform

3.1.1 Usage scenarios related with data activities.

For the data onboarding and management activities, partners of the project expect the platform to support onboarding and saving datasets of different types (tabular and image data). They also expect to be able to transform and use different version of data for different experiments. In addition, partners expect to be in position to apply different quality and validation approaches on top of the data stored in the experimentation platform. Finally, it is expected access control policies to be applicable securing authorization on the usage of the data layer.

3.1.2 Usage scenarios related with developing machine learning pipelines.

Partners of the project expect to be in position to utilize scalable compute resources when it comes on developing and executing machine learning pipelines and also it is expected the support of GPUs that accelerate the machine learning training operations. Additionally, it is expected that partners will be in position to upload and execute source code related with their machine learning without interfering to the execution of experiments originated by other partners. Furthermore, it is expected that the experimentation platform will offer possibilities to its users to track and version and audit the results (artifacts) of the experiment sessions.

3.1.3 Usage scenarios related to the machine learning deployment, operations and maintenance activities

Partners expect to use the experimentation platform for the end-to-end life cycle of machine learning models. This means that the experimentation platform needs to support the deployment of prediction / inference services and the continuous operation, monitoring and maintenance of them. Capabilities for retraining models and deployments of new model versions to production for maintaining a good quality of models are expected to be met.

3.1.4 Usage scenarios related with accessing and deploying the experimentation platform

A final usage scenario described for the experimentation platform is that it should be possible for the experimentation platform to be both deployed to local machines and to cluster of servers that could be located in cloud or on premises.

3.2 Architectural view and considerations

For addressing the usage scenarios and the minimum requirements that were set to the experimentation platform a basic high-level architecture was described. This architecture describes five basic layers:

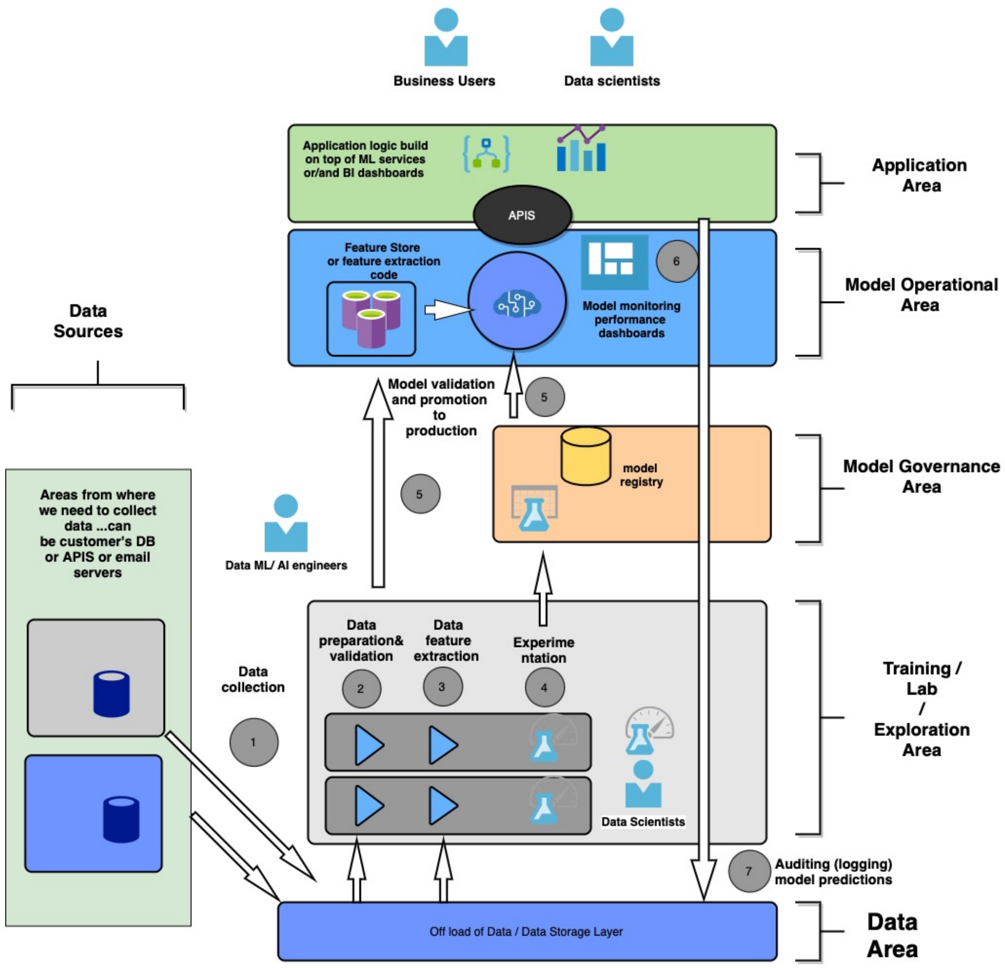

Figure 3: Basic architectural layers in the experimentation platform. Note this picture visualizes and the Application area from where the predictions of ML models are accessed and used. This area is not considered to be part of the experimentation platform thus it is not described in the following section. Our model operation area should provide interfaces to the Application area for retrieving predictions.

1. The data layer

Purpose of the data layer is to provide to the users of the platform a layer where they can store, access, transform, version and expose data for the operation of the machine learning pipelines.

2. The machine learning developing / engineering layer

Purpose of the ML developing layer is to provide to the users of the platform a layer where they can execute the machine learning training pipelines. This layer should provide means for scalable compute resources, for utilizing a variety of machine learning frameworks without creating conflicts on the different library versions, means for tracking, comparing and storing experiment results, means for applying automation on the execution of the training pipelines.

3. The Machine Learning governance area

Purpose of the ML governance area is to provide a solution to the users of our platform to control and manage the ML pipelines that operate in production. Through the machine learning governance area, a user of the platform will be in position to evaluate ML training operations, to decide when to deploy a new model version to production or to staging environments, to ensure that all the ML quality assessment operations have been completed

4. The machine learning operational area

Purpose of the machine learning operational area is to provide to the users of our platform a way to serve models and to monitor their execution. In this layer, the models should be exposed in a way that their predictions should be reachable form the business / application layer. This means that the model operational layer should provide capabilities to expose models behind APIs for the purposes of on-line ML use cases. In addition, this layer should provide capabilities to execute models periodically and expose the predictions in data storages from where business applications can access them. Executing models periodically and providing predictions usually take place for the purposes of batch use cases.

5. The access control layer

Purpose of the access control layer is to provide the means for managing the accessibility of the resources of our platforms to the users. Resources can be datasets, machine learning artifacts, rest APIs, source code and so on.

All of the above layers should operate on top of the infrastructure of the experimentation platform and different MLOps tools and software components can be introduced for addressing the requirements that its particular layer brings.

3.3 Software components and technology options

For addressing the requirements described by the usage scenarios and for establishing the architecture described, partners in the project decided to suggest particular software components and MLOps tools. The suggested tools are well-established open-source technologies that are maintained by large communities of developers and enterprises.

The following table presents the MLOps tool that partners in IML4E project decided to utilize for the experimentation platform. The table is an outcome of the Berlin meeting that took place at June 2022. The columns in the first row in the table describe the need (feature) that the suggested tool cover. The columns in the second row in the table describe the suggested MLOps tool that cover the need. **X** in the table is used for positive vote towards the usage of a tool, **IH** (In House) is used to describe the fact that a partner will use an in house-built solution for the specific need, **- (or empty box)** is used to describe that a partner is not in need for such a feature in their use-case**.** In regards on selecting a tool for feature store, partners are not in position to suggest a tool yet, and the input from work package 2 (WP2) is going to contribute to the selection.

Case study	Experiment tracking	Model Registry	Pipeline automation	Featrue Store	Model Serving (Online)	Visualization tool
	MLFlow	MLFlow	Kubeflow pipelines *	?	KServe	Grafana , Prometheus
Siemens	X	X	X	Feast	X	X
Granlund	X	X	X	No suggestion	X	X
Basware	- (added complexicity)	X	X	IH (MongoDB and S3)	IH (In house solution)	IH
Reaktor	X	X	X	IH	X	X
Vitarex	X	X	To be decided if needed	No suggestion	offline model serving	perform batch analysis
* we have kubernetes as our basis						

Figure 4: Table of MLOps tools to be utilized in the IML4E project for the purposes of the experimentation platform. Source of the figure is an Excel file maintained by the IML4E project.

As it is described later in the Summary and future work section, partner's aim to evaluate the feasibility of those tools for the establishment of an experimentation platform by introducing ML use cases on top them.

Suggested Software components and tools:

1. Kubernetes

It was decided to introduce Kubernetes for the deployment of containerized solutions. In addition, Kubernetes meets the requirements for having a scalable compute layer that can include CPU as well as GPU resources. Finally, Kubernetes can run on local machines as well as in clusters of machines and it is support in the majority of cloud providers. (Kubernetes, 2022)

2. Kubeflow pipelines

It was decided to use Kubeflow pipelines for addressing the needs of "the machine learning developing / engineering layer". Kubeflow pipelines enable the orchestration and automation of ML pipelines on top of Kubernetes. They help on managing dependencies between the different execution steps in an ML pipeline and they enable the execution of multiple ML pipelines in parallel. Kubeflow runs on top of Kubernetes, and it was originated from Google. (Kubeflow pipelines, 2022)

3. MLFlow

It was decided to use MLflow for the purposes of managing the needs of "the Machine Learning governance area". MLflow supports metadata tracking of the different artifacts generated during the execution of one ML pipeline. In addition, it supports comparison of different ML experiments, and it also supports a layer for governance regarding the state of the models [state can be "None', 'Archived', 'Staging', or 'Production'].

MLFlow can run on top of Kubernetes and it was originated from an company named Databricks and now is part of the Linux Foundation.(MLFlow 2022)

4. KServe

It was decided to use KServe for the purposes of managing the needs of "the Machine Learning operational area". KServe supports the deployment of ML models behind REST-APIs, and it also supports predictions for batch data. KServe runs on top of Kubernetes and it is originated from the same project as Kubeflow pipelines. In addition, KServe integrates well with Grafana and Prometheus tools that we are using for our monitoring purposes. (KServe 2022)

5. Grafana and Prometheus

It was decided to use Grafana and Prometheus for the purposes of monitoring and visualizing the operations of the ML learning models. This technology stack is very common in monitoring modern solutions and most of the enterprises have already introduced those tools in their technology stack.

6. GitLab

It was decided to use Gitlab for the purposes of tracking the source code and for the purposes of enabling continuous delivery of code.

With the above software components, we meet the basic requirements of our experimentation platform. Requirements around versioning, automating, monitoring and deploying models**. For addressing more requirements such as automation in the ML development process (AutoML) and hyper parameter tuning more tools can be introduced.** However, in the first iteration of the project we aim to introduce the basic tools that meet the basic MLOps requirements. As the project advances and by integrating our experimentation to use cases, there is the possibility of introducing new technologies and tools to our MLOps tool chain.

Availability Considerations and Requirements

The experimentation platform should aim 99% ("two nines") availability for the beginning of the project.
This availability translates to 14.40 minutes downtime per day. Or 21.9 hours per quarter.

4 Training Platform Requirements in ML R&D And AI Utilization

4.1 Usage Scenarios and Use Cases

Most probable scenario is that during the IML4E project there will be several separate sets of training material related both to concepts, ways-of-working, and technology. The materials will be provided by several parties independently, and in incremental steps. The following content is expected

1. Text-based learning material (slides, blog posts, handbooks, "playbooks") produced in IML4E
2. Video lectures and how to-do videos
3. Examples (instructions, code, and data) for hands-on experiments with the experimentation platform
4. Curated, self-paced learning paths on collected IML4E training material, and external content.

4.2 Content Generation Considerations and Requirements

Most of the content is expected to be generated by the usual text, and other media generation software. There should be appropriate guidelines available for preferred formats to ensure access to most users. Also, style guides and sufficient logos and possibly other project related graphics material repositories are needed.

4.3 Availability Considerations and Requirements

In most cases, the training material is expected to be publicly available. It requires further consideration whether access to the environment should still require registration. If part of the material is restricted to limited audience or partnerships, there must be access rights management.

Availability of the project training material after the project, length of the availability, and responsibility for organizing this, and funding must be negotiated and agreed.

4.4 Architectural Considerations and Requirements

The architectural requirements depend much on the type and amount of content that is created during the project. As a starting point, the web pages of the https://iml4e.org/ is used to collect pointers to the material, either under "*Resources*", or a new separate *Training* section. In the beginning, the different training material providers will host their material as they find appropriate, e.g., in cloud services, and this material is linked to project web pages. The material providers also can handle access rights is necessary.

5 Technical Requirements for operating the Experimentation and Training Platform(S)

In this section technical requirements are presented, for hosting and operating both the Experimentation and the Training platform. The technical requirements aim to describe which should be the technical characteristics of the systems that are going to host the experimentation platform and the corresponding educational material in the context of IML4E project. It is reminded that the usage scenarios of those platforms and their requirements are described in section 4 and 5.

In the plenary meeting end of March 2022, it was agreed that for the platforms a comprehensive approach will be followed. That means that all platforms will be, in a first version, integrated in a single platform. If, at any later stage the decision is taken to separate the platforms into a training platform / portal, and an experimentation platform there is no restriction to these endeavors. But, in order to make all of these platforms available the effort to achieve this within the course of this project seems quite high. This together with the fact that writing detailed comprehensive requirements for such platforms is exceeding the scope of this report, led us to the idea of introducing a minimal viable product (MVP). This MVP is still a challenge but much more realistic to achieve.

5.1 Technical Requirements for operating an Experimentation Platform

For operating a full-scale experimentation platform we would need to provision and maintain significant hosting infrastructure; investing time, human experts and budget resources to this. Thus, it was decided to suggest a minimum version (Minimum Valuable Product-MVP) of the hosting/operating environment that would be utilized by our experimentation platform**. This MVP technical platform will be materialized by local machines and private computers. The MLOps tools and software components, described in section 4, are going to be versioned and maintained in a Git repository together with the installation instructions for local deployment and operation of the experimentation platform.**

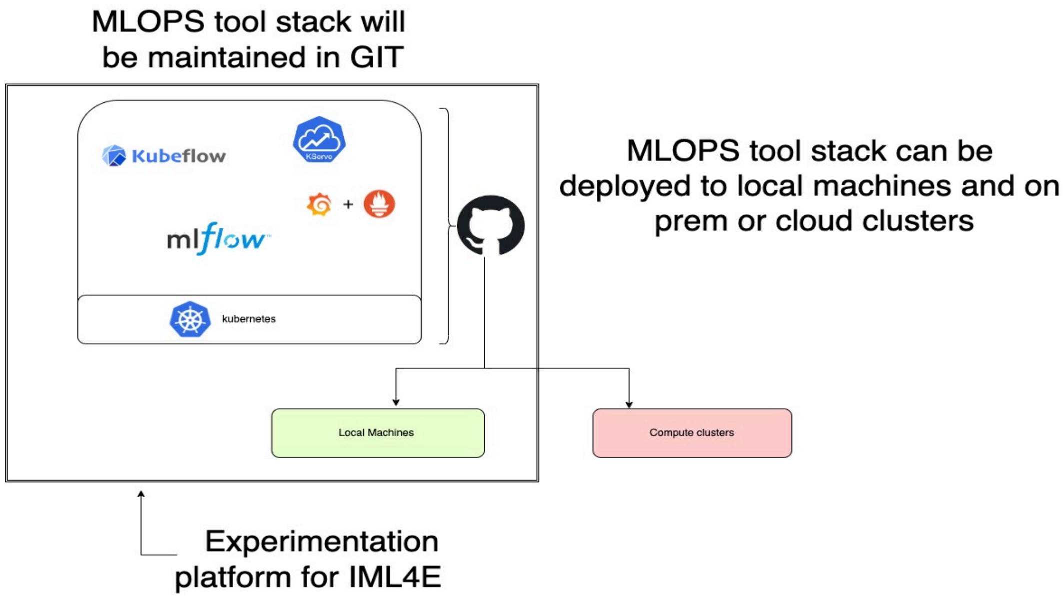

Figure 5 : Visualizing the experimentation platform's MVP approach. The software stack will be maintained in GIT and the platform could be deployed to several target environments. For the purposes of IML4E we have selected to support the deployment to local machines only.

As described in section 4, a vital software component of our experimentation platform would be the operational layer of it. The experimentation platform - even on local machines - will operate on a container-based orchestration solution (Kubernetes). **The introduction of that technology will ensure that our solution will scale and be portable to more advanced technical setups with multiple compute and data resources hosted in cloud or in premises.**

In the following paragraph we briefly describe the resources that we should have in place for a full scale ML/AI experimentation platform:

- Servers to host the platform (Linux) (central or distributed)
- Administrator(s) to administer the servers
- Setting up of layers with different access rights, compare **Fehler! Verweisquelle konnte nicht gefunden werden.**, i.e., several servers
    - User management to handle access control
- Responsible person(s) to monitor the platform(s):
- Data storage and management of data sets
- Network connectivity and availability to access the platform

In our MVP approach we do not need to maintain and provision the above resources for the experimentation platform. On the contrary, we are only going to maintain a GIT repository that includes

- Installation instructions on how to operate and execute our MLOps tool chain in local machines
- Installation instructions on how to run and execute Kubernetes in local machines
- A project template that will simulate the execution of an ML pipeline utilization the defined MLOps toolchain and common software components.

5.2 Technical Requirements for operating Training Platform

For hosting the training and educational material a similar MVP approach it is decided to be followed as it is decided for the experimentation platform. It is decided that the training material will be stored and maintained to a GIT – like platform. In the GIT platform the training material will be described with the necessary README files, and capabilities for referencing articles and emending videos and documentation should be supported. Uploading and maintaining large files in the GIT platform will be avoided, and instead references to those large files are going to be stored.

Finally, it was decided that the same GIT platform that will host the experimentation platform it will also host and the training material/ platform.

5.3 Requirements for Making the Platform(S) Available

As described above a GIT – like platform will be used for hosting the experimentation and training platform. There are several GIT – like platforms offering paid hosting as well as free use if hosting is done externally. It is therefore necessary that one of the partners take over the role of major host and installs, maintains the GIT server and provides each of the partner organizations that will contribute to the platform with access credentials. Each partner organization should provide at least one contact person that will have access rights to the platform and each organization is responsible for uploading and contributing to the experimentation and training platform with respect to the rules and directions from the IML4E project.

In regards to the lifecycle of the git platform, it is to be negotiated what will happen after the end of the project.

6 Summary

This document discusses the requirements and the characteristics of the experimentation and training platforms (*chapters* 4 and 5). It introduces established technologies for materializing the platforms (section4.2) and suggests a way forward for developing those platforms as minimum viable products (MVP) that easily can scale to Enterprise level solutions after provisioning the right technical infrastructure and components. The part of introducing the MVP is discussed in chapter 6.

The motivation for introducing an MVP deliverable for the experimentation and education platform derives from the fact that the maintenance, portability and usage of the MVP platform is considerably easier than maintaining a full-scale ML/AI platform (chapter 6). At the same time the requirements that are not met by the MVP platform are not critical for demonstrating and operating basic MLOps practices and frameworks. We decided not to cover requirements related with handling and introducing GPUs, big datasets and compute clusters.

7 Future work

Next step for WP4 would be to materialize the experimentation and education platform in a code repository (e.g., GitLab). Additional next step will be the integration of a project's use case on top of the described MLOps toolchain. The integration of a use case will help in evaluating our tool suggestions for the experimentation platform and also will generate educational and training material. Finally, further, work for the WP4 will be also to define processes and a framework describing the execution of ML pipelines on top of the experimentation platform.

References

Lwakatare, L. E., Crnkovic, I., & Bosch, J. (2020). DevOps for AI–Challenges in Development of AI-enabled Applications. In 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM) (pp. 1-6). IEEE.

Muiruri, D; Lwakatare, L,E; Nurminen, K, J; Mikkonen, T (2021): Practices and Infrastructures for ML Systems – An Interview Study in Finnish Organizations. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.16939192.v2

Google LLC, 2020. MLOps: Continuous delivery and automation pipelines in machine learning. [Online] Available at: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning
[Accessed 5 June 2022].

Ozkaya, I., 2020. What is really different in engineering ai-enabled systems?. IEEE Software, 37(4), pp.3-6.

Amershi, S et al., "Software Engineering for Machine Learning: A Case Study," 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 2019, pp. 291-300, doi: 10.1109/ICSE-SEIP.2019.00042.

Kubernetes io, Avaliable at https://kubernetes.io/ [Accessed: 2 June2022].

MLFlow , Avaliable at http://mlflow.org/https://kubernetes.io/ (Accessed: 2 June 2022].

Kubeflow Pipelines, Avaliable at https://www.kubeflow.org/docs/components/pipelines/sdk/build-pipeline/ https://kubernetes.io/ [Accessed: 2 June 2022].

KServe, Avaliable at https://www.kubeflow.org/docs/external-add-ons/kserve/kserve/ https://kubernetes.io/ [Accessed: 5 June 2022].