2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC) | 979-8-3503-2697-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/COMPSAC57700.2023.00185

Linking Team-level and Organization-level Governance in Machine Learning Operations through Explainable AI and Responsible AI Connector

Elie Neghawi
Electrical and Computer Engineering
Concordia University
Montreal, Canada
e_negh@encs.concordia.ca

Zerui Wang
Electrical and Computer Engineering
Concordia University
Montreal, Canada
zerui.wang@concordia.ca

Jun Huang
Electrical and Computer Engineering
Concordia University
Montreal, Canada
jun.huang@concordia.ca

Yan Liu
Electrical and Computer Engineering
Concordia University
Montreal, Canada
yan.liu@concordia.ca

***Abstract*—The adoption of AI systems has been widely used across multiple industry domains at an alerting rate without focusing on its ethical concerns. In order to address those concerns, an increasing number of AI ethics frameworks have been suggested recently, which focus on the algorithmic level rather than the systems level. Nonetheless, some system-level approaches mostly cover a single-level governance pattern of the system components in the entire software design life cycle. However, the need to go beyond the single-level system design AI ethics frameworks to allow not only a better responsible-AI-by-design but also a trustworthy process pattern that abstracts and links the underlying layers of responsible AI on every level. This paper illustrates a principal-to-practice guide of the multi-level governance within organizations across the globe for AI ethics frameworks. We outline the main gap areas in organizations for AI ethics frameworks. Consecutively, we propose a multi-level governance pattern for responsible AI systems within organizations which is participatory, iterative, flexible and operable that targets those main gap areas. Finally, to assist practitioners in applying the multi-level governance AI in organizations and its impact on the industry level, we will translate it into effective and responsible AI practices using a case study.**

***Index Terms*—AI, AI ethics, trustworthy AI, XAI, AIMLOps, AIOps, software engineering, software architecture, pattern, best practice**

I. Introduction

Artificial Intelligence (AI) reshaped our lives and helped people make better predictions and more informed and wise decisions. However, these high techs are still in their infancy, and there remains much promise for AI to promote innovation and address global challenges people face.

Consecutively, ethical concerns and anxieties are fuelling around AI [1]. Many enquiries exist on the trustworthiness and adoption of AI systems, including concerns about exacerbating inequality, the digital divide, climate change and market concentration. Additionally, there are concerns that the use of AI may compromise human rights and values such as privacy. To address these concerns and ensure the responsible development and use of AI, a collaborative effort involving multiple stakeholders and international cooperation issued guidelines and ethical principles. Despite the creation of ethical guidelines for AI development inside organizations, it can be challenging for developers to apply these principles in practical situations. These principles are often abstract and may not provide clear direction for specific implementation [2]. Therefore, more specific and actionable guidelines are needed to assist developers in implementing ethical considerations in their AI systems. XAI, or Explainable Artificial Intelligence, can provide clear explanations for the decision-making process of AI systems. This understanding can help build trust and confidence in the system and its developers. In addition to transparency and accountability, XAI can also help ensure that AI systems are non-discriminatory. It is important to bridge the gap between ethical principles and the algorithms used in AI systems to ensure responsible development. However, the architecture of an AI ecosystem consists of three layers: AI software supply chain, AI system, and operation infrastructure. It is challenging to show the contribution of each.

One proposed work is Responsible AI (RAI) Pattern Catalogue [3], which takes a pattern-oriented approach to promoting RAI in practice. Instead of solely focusing on ethical principles or AI algorithms, this catalogue focuses on design patterns so that practitioners can apply them to ensure their AI systems are responsible throughout the software development process. The catalogue is organized into three categories: 1) governance patterns to establish multi-level governance, 2) process patterns to estab-

979-8-3503-2697-0/23/$31.00 ©2023 IEEE
DOI 10.1109/COMPSAC57700.2023.00185

lish trustworthy development processes, and 3) product patterns to integrate responsive design into AI systems. In addition, it focuses on all aspects of the ecosystem (industry-level, organization-level and team-level) without the planning of the design and the development tools to support the navigation and utilization of the RAI pattern catalogue.

In this paper, we take a different approach by focusing on the organization-level patterns at the system level rather than just the ethical principles or AI algorithms. This approach aims to integrate responsible design in organizations into final AI products by looking at the Machine Learning Operations (MLOps) in a bigger picture and the design patterns that reshape the system. This is done by bridging the gap between the organization-level and team-level and facilitating navigation using MLOps. We start by looking at the main two levels of an organization in addition to the team-level and examine the currently available methods [6]–[10]. Then we make the links on where those methods meet and create the best practices using the multi-level governance patterns at the organization-level. The overarching research question that has guided this study is:

> What is the multi-level governance pattern principle-to-practice proposed for responsible AI systems to bridge the gap between team-level and organization-level using MLOps?

The main contributions of this paper are as follows:

- Find the link between team-level governance patterns and organization-level patterns.
- Suggest navigation and utilization of team-level governance patterns with the organization-level patterns.
- Explore a case study that suits this principle-to-practice multi-level governance pattern.

II. Related Work

The issue of creating AI that is ethically accountable has garnered a great deal of interest among both industrial and academic communities. To promote ethical AI practices, a multitude of AI ethics principles and guidelines numbering around 100 have been established by various entities, including governments, companies, and organizations [12]. However, these guidelines often need to be more general and theoretical for individuals involved in the implementation of AI systems to apply in real-world scenarios.

There has been a concerted effort in AI to address the challenges of RAI. One approach that has gained traction is the development of algorithm-level solutions. These solutions are designed to address specific aspects of the numerous high-level AI ethics principles and guidelines that various entities have established. By focusing on a subset of the principles, these algorithmic solutions aim to bring concrete and practical approaches to address some ethical concerns related to AI. One approach developers used is limiting user access and preventing reverse engineering or modifications to the system design. Rather than providing full access to AI systems by running them locally, it is recommended to offer AI services through cloud-based platforms and manage interactions through APIs [11]. As an illustration, access to OpenAI's language model GPT-3 is limited to approved users who can only integrate it into their AI systems via API. Another example is Google Vision AI's facial recognition feature, which is limited to a select few celebrities and only accessible through API. Despite these efforts, there have been instances where the algorithm has been exposed to the outside without proper internal review and verification, leading to potential issues with the responsible use of AI.

However, it is important to note that these algorithm-level solutions are just one part of the larger picture of RAI. There may need to be more than just implementing them to address all the ethical concerns related to AI, as the principles are often complex and multifaceted. It requires a collaborative effort between researchers, developers, policymakers, and other stakeholders (board members, executives, and managers) to ensure that AI is developed and used ethically and responsibly.

III. Methodology

In order to build up the links of the multi-level governance for RAI systems within organizations, we first evaluate the available methods at the team and organizational level [3] to understand their strengths and limitations. We then identify the gaps that provided opportunities for improvement. As shown in Figure 1, the hierarchy of the organization and team-level stakeholders in the industry is depicted on the left side of the illustration, providing a visual representation of the various levels of responsibility and decision-making within the industry. The right side of the figure displays the current methods available, which are being utilized to support the operations and processes of the stakeholders.

The illustration provides a comprehensive overview of the stakeholders involved and the methods being utilized, offering insight into the strengths and limitations of the current methods. In addition, the use of XAI and RAI connectors, as shown in the illustration, can further optimize the operation of the current methods and support the efforts of the stakeholders. Utilizing these connectors can provide a more comprehensive and user-friendly experience, leading to improved outcomes and increased success for the organization and its teams.

Furthermore, we evaluated an examination of MLOps technologies and tools for each stage of the project pipeline and the roles involved [14]. In this examination, we identified the weakness of the method being used as the absence of XAI and RAI. The lack of XAI and RAI in the method being used can result in unintended consequences and decreased trust in the system. Therefore,

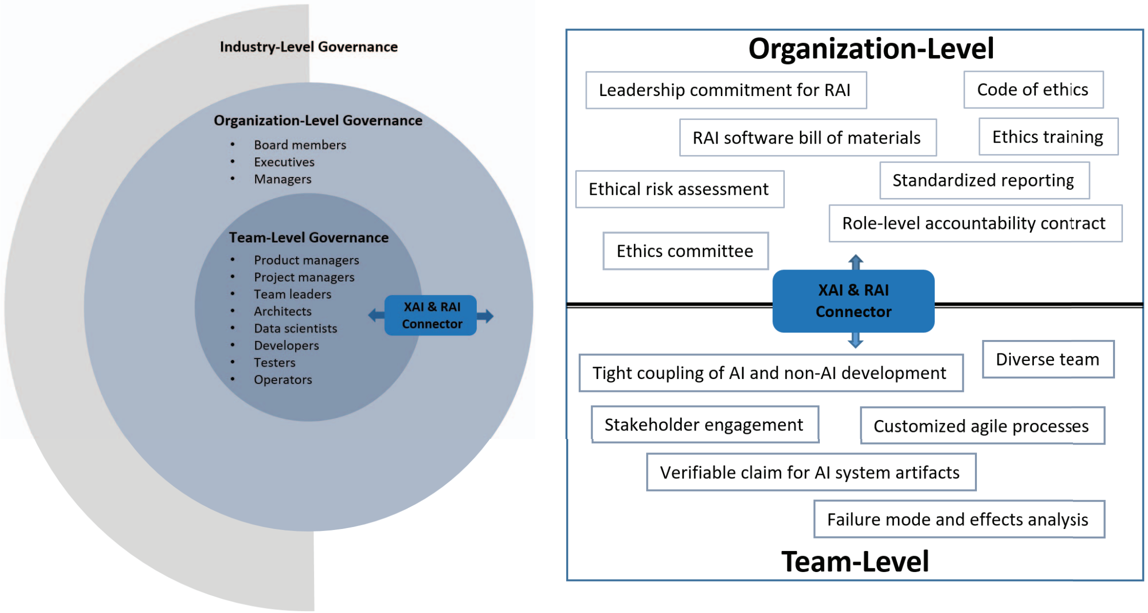

Fig. 1. A Deep Dive into the MLOps Workflow with the XRc Phase: Understanding Key Actor Roles and Responsibilities

it is essential to consider incorporating these elements into any machine learning project to ensure accountability and transparency. To the best of our knowledge, there is no standard for implementing the multi-level governance pattern for RAI with XAI in MLOps.

The XAI and RAI connector (XRc) can play a crucial role in connecting team-level governance to organization-level governance implementation in MLOps. By providing clear and understandable explanations for the decisions made by machine learning models, XAI helps to increase transparency and accountability at the team level. This can be especially important in complex projects involving multiple stakeholders and team members. RAI, on the other hand, helps to ensure that ethical and moral considerations are taken into account throughout the entire MLOps pipeline. This can involve creating policies and guidelines for RAI and conducting risk assessments and impact evaluations. By incorporating RAI into MLOps, organizations can ensure that their use of AI aligns with their values and meets regulatory requirements.

By introducing XRc into the MLOps, organizations can bridge the gap between team-level governance and organization-level governance implementation in MLOps. This helps to ensure that AI systems are used responsibly and ethically while providing a clear and transparent explanation of their decision-making process.

IV. Background on MLOps Workflow Stages

Constructing a machine learning pipeline can be a challenging endeavour. The pipeline is often constructed incrementally with the assistance of tools that have limited integration capabilities. MLOps seeks to streamline this process by automating the pipeline. It combines machine learning, data engineering, and DevOps practices, essentially streamlining and accelerating the operationalization of an ML model (including building, testing, and releasing) by incorporating DevOps practices into the process. Determining which stage should be executed by which actor in the MLOps pipeline is complex and often requires multiple iterations to arrive at a suitable solution. However, four major stages have been identified by examining multiple studies. Subsequently, we will outline each component in detail for that particular stage.

a) Data Management Phase: It can be challenging due to domain-specific limitations [53] that affect relationships between attributes, historical records' accuracy, and state transitions [30]. Domain experts are crucial in ensuring that data models align with project goals and KPIs. They validate potential data and machine learning models to meet project requirements. Some organizations employ Data Stewardship to oversee data quality management and governance, with defined roles such as chief, business, technical data stewards, and a data quality board [52].

b) ML Preparation Phase: This set of functions in the ML preparation stage deals with classic ML preprocessing tasks. Data quality is important and ensured by various roles with the help of data engineers and stewards. Implementing the ML model requires collaboration between data scientists, domain experts, and those responsible for defining the problem within the domain [42]. In summary, here are the functions in the ML preparation phase:

- **ML Data Acquisition**: The ML pipeline is fed with relevant data based on the prior declared data management plan and selection by the data engineer.

- **Data Cleaning and Labeling**: The input data is cleaned and labeled for ML operations with the help of data scientists and domain experts [30].
- **Data Versioning**: The separation of test, training, and validation data sets is crucial for the success of ML models and is achieved through data versioning.

c) ML Training Phase: The role of data scientists is crucial in the ML Pipeline Phase. They ensure the flexibility, scalability, and proper technology selection of the ML pipeline while also working to improve model performance. They select the ML pipeline structure, algorithms, and hyperparameters through model versioning and validation and are the main users in data processing for big data projects. AutoML techniques [44] and tools support data scientists and domain experts in efficiently selecting the ML pipeline and training the model. The process includes feature preprocessing automation, model selection, and hyperparameter optimization. To sum up, the following functions are performed in the ML preparation phase:

- **Pipeline Structure Search**: The structure of an ML pipeline depends on the type of data (structured or unstructured) and the technique used to solve the problem (supervised, unsupervised, or semi-supervised learning). The specific performance metrics to be used must also be defined based on the specific domain-specific requirements of the problem being solved.
- **Algorithm and Hyperparameter Selection**: The choice of the most suitable ML algorithm for a problem is made by data scientists. The algorithm's performance can be improved by adjusting its hyperparameters, such as the number of layers in a neural network [36]. However, this process can be time-consuming and complex which addressed by AutoML in [44].
- **Model Versioning**: It is a way to keep track of the interdependencies between an ML model, its data, framework, and modelling procedure. It is important to revert to previous models if there is a problem in production and then deploy the correct version at the right time. Model versioning increases accountability and is essential for managing complex ML models.

d) Deployment Phase: The deployment stage is a pivotal moment in the MLOps process. During this phase, software engineers are responsible for incorporating the approved models into the corresponding applications and ensuring the smooth operation of the entire system. To maintain this stability, MLOps engineers must continuously monitor the model, the application as a whole, and the data being used [42]. Another key player in this phase is the DevOps engineer, who is responsible for constructing, testing, and deploying the functioning system. In general, it is characterized by the following tasks:

- Integration of validated models into the relevant applications by software engineers.
- Maintenance of the operational stability of the entire system by MLOps engineers through continuous monitoring of the model, application, and data.
- Construction, testing, and deployment of the functioning system by DevOps engineers.

V. XAI and RAI Connector(XRc)

The integration of XRc into the MLOps pipeline may come with added overhead. However, it proves to be a valuable addition to the process as a whole. The addition of XRc not only reduces the risk of failure in RAI but also promotes efficiency by allowing for early detection of any problems with implementing organizational-level governance. This helps to avoid duplicating efforts and ensures that the RAI is being developed effectively. As shown in Figure 2, XRc has been inserted between the ML pipeline and ML deployment phases to analyze the changes before migrating into the application and incorporating the organization-level governance into the process. Let us now delve into the sub-phases of XRc.

A. Model Type Identification

Both dynamic and static identification methods can be used to identify the type of machine learning model, with dynamic methods involving examination of the model's API or performance and static methods involving examination of the code and architecture used to implement the model.

a) Static Identification Method: In code, the type of machine learning model can be identified by examining the architecture, algorithms, and libraries used to implement the model. Understanding the architecture of the model, such as the number of hidden layers or the presence of decision trees, can give a good indication of the type of machine learning model used. Additionally, many machine learning libraries provide pre-built models with clear documentation that specify the type of model being used. The documentation for these libraries usually clearly states the type of model being used. For example, in the scikit-learn library [37], the use of the 'LogisticRegression' class for logistic regression, which is a supervised learning algorithm, or the 'KMeans' class for k-means clustering, which is an unsupervised learning algorithm.

b) Dynamic Identification Method: There are two main ways to detect the type of machine learning model dynamically:

- **Examination of Model API and Function**: This involves looking at the functions that the model exposes, such as the 'predict' function, and determining the type of model based on the inputs and outputs of the function.
- **Examination of Model Performance**: This involves evaluating the model's performance on a known

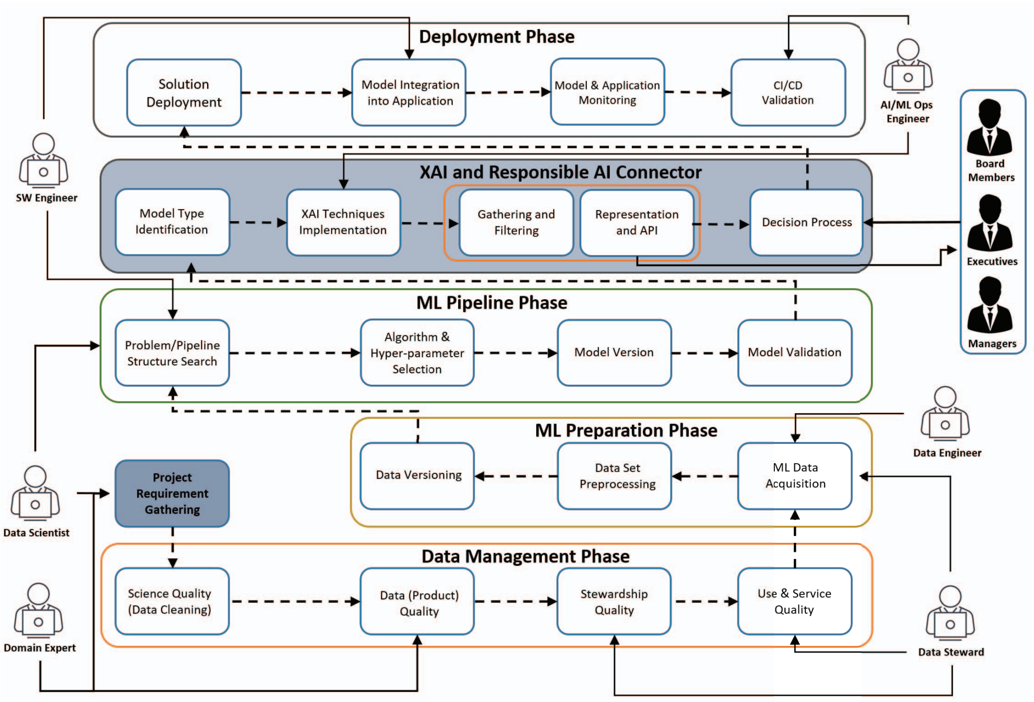

Fig. 2. The XAI and RAI connector (XRc) to bridge explainable and responsible operations with machine learning life cycle

dataset and determining the type of model based on the performance metrics and results obtained.

The choice of dynamic method depends on the specific requirements of the use case. For example, examining the API or functions can be easiest if they are accessible, whereas evaluating the model's performance on a known dataset may be the only option if there is no API or function access.

B. XAI techniques Implementation

XAI techniques are implemented by AI/MLOps engineers considering the ML algorithm and the different audiences. It may require a trade-off between technical detail and simplicity, transparency and fairness, and other factors.

a) Audience Evaluation: While the audiences is mainly at the organization level, AI/MLOps Engineers can categories their audience into two:

- End users require simple and understandable explanations of the decisions made by a machine learning model. XAI techniques for this audience include rule-based systems, decision trees, or prototype-based explanations.
- Regulators require explanations of the decisions made by a machine learning model to ensure compliance with regulations and ethical standards. Model-agnostic techniques like Local Interpretable Model-Agnostic Explanation (LIME) [54] or SHapley Additive exPlanation (SHAP) [55] can provide explanations for the predictions made by any machine learning model.

b) ML Algorithm Evaluation: It is not necessary for an AI/MLOps engineer to have a deep understanding of the specific ML algorithm in order to choose an XAI method. However, having a general understanding of the ML algorithm and its strengths and weaknesses can help choose an appropriate XAI method. In addition, since some XAI methods can be used and integrated together, AI/MLOps engineers have to consider the broad range of XAI techniques. Those methods can be categorized into the following categories [13]: Model-Agnostic Techniques: Model-agnostic XAI techniques can be applied to any machine learning model, regardless of the underlying algorithm or architecture. Examples of model-agnostic XAI techniques include LIME and SHAP.

- **Model-Specific Techniques**: Model-specific XAI techniques [13] are designed specifically for a particular type of machine learning model, such as decision trees or neural networks. Examples of model-specific XAI techniques include saliency maps for neural networks and decision trees for decision trees.
- **Post-Hoc Techniques**: Post-hoc XAI techniques [13] are techniques that are applied to a trained machine learning model after it has been trained to explain its decisions and actions. Examples of post-hoc XAI techniques include LIME, SHAP, and saliency maps [47].
- **Integrative Techniques**: Integrative XAI tech-

niques involve integrating XAI into the training process of a machine learning model so that explanations can be generated as part of the model's regular operation.

These categories are not mutually exclusive, and some XAI techniques may fall into multiple categories. The choice of XAI technique will depend on the specific requirements of the use case and the type of model being used.

C. Gathering and Filtering

Gathering and filtering XAI methods involves selecting a subset of XAI methods that are appropriate for a specific use case and then integrating and combining those methods to provide a more comprehensive and effective explanation of the decisions made by a machine learning model. To effectively utilize XAI methods for a specific use case, here are the steps to gather and filter XAI methods:

- **Define the requirements**: AI/MLOps engineers should start by defining the requirements based on the audience they are trying to target and the type of model being used. This will help to determine which XAI methods are most appropriate for the use case.
- **Gather XAI methods**: Next, gather a set of XAI methods that are appropriate and generated by the XAI technique Implementation. This may require back and forth to add and remove certain XAI techniques.
- **Filter XAI methods**: Once a set of XAI methods is gathered, filter the methods based on the specific requirements of the use case. Consider factors such as the complexity of the method, the amount of computational resources required, and the amount of technical detail that is appropriate for the audience.
- **Combine XAI methods**: After filtering the XAI methods, MLOps Engineers might consider combining methods to provide a more comprehensive and effective explanation of the decisions made by the machine learning model. They can use techniques such as multi-modal explanations, ensemble explanations, or hybrid explanations to combine XAI methods.

D. Representation and API

XAI representation refers to the format in which the explanations generated by XAI techniques are presented to the user. The representation can be in the form of text, visualizations, or other forms of data, and it depends on the specific requirements of the use case and the target audience.

An XAI API, or Application Programming Interface, is a set of protocols and tools for building software applications that provide access to XAI capabilities. An XAI API allows developers to easily integrate XAI techniques into their applications and provide explanations for the decisions made by machine learning models. For example, an XAI API might provide functions that allow developers to generate explanations for specific predictions made by a machine learning model or to visualize the model's decision-making process. The XAI API might also provide a set of data structures and protocols for representing the explanations generated by XAI techniques, such as text, visualizations, or other forms of data. Here are some ways to make the XAI API interactive:

- **Visualizations**: Use interactive visualizations, such as heat maps, bar charts, and scatter plots, to help stakeholders understand the explanations generated by the XAI API. This can include interactive visualizations of the model's behavior, such as feature importance, or visualizations of individual predictions, such as decision trees.
- **User Input**: Allow stakeholders to provide input to the XAI API, such as selecting specific predictions to explain or adjusting the parameters of the explanations. This can help stakeholders better understand the explanations generated by the API and tailor the explanations to their specific needs.
- **Dynamic Explanations**: Provide dynamic explanations that change based on user input or other factors, such as the type of model being used or the specific prediction being explained. This can help stakeholders better understand the API's explanations and see how changes in the model or input data affect the explanations.
- **Feedback Mechanisms**: Provide feedback mechanisms that allow stakeholders to provide feedback on the explanations generated by the XAI API. This can include simple feedback forms, or more complex mechanisms, such as a feedback rating system. This can help organizations improve the quality of the explanations generated by the API and better meet the stakeholders' needs.

In a nutshell, XAI representation refers to the format in which XAI explanations are presented to the user. At the same time, an XAI API provides a set of protocols and tools which integrate XAI techniques into software applications and explain the decisions made by machine learning models. The choice of XAI representation and API will depend on the specific requirements of the use case and the target audience.

E. Decision Process

Incorporating an XAI API into the decision-making process at the organizational level can help stakeholders make informed decisions about whether to proceed with or stop an ML algorithm based on the transparency and fairness of the model. To make the XAI API interactive, SW Engineers and MLOps can add features that allow stakeholders to interact with the explanations generated by the API.

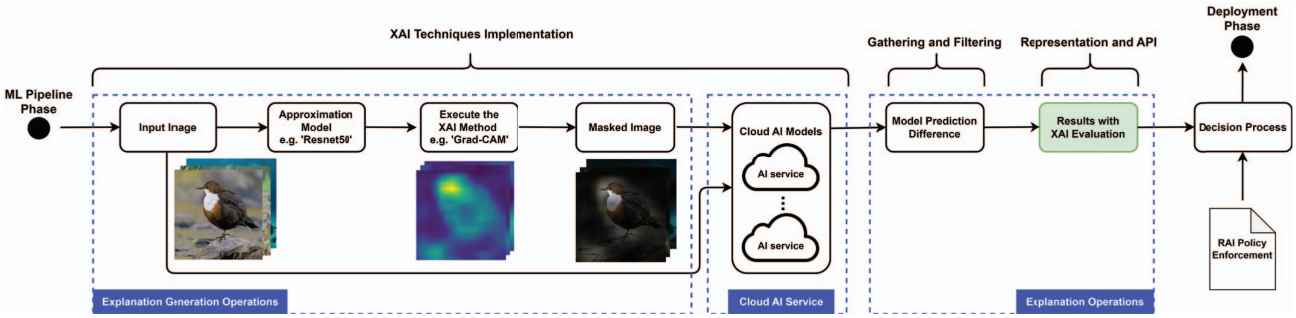

Fig. 3. An XAI and RAI Connector (XRc) implementation case study for cloud-based image classification AI model

VI. Example Conceptual Scenario

We elaborate on an XAI service for cloud AI and develop its XRc implementation as an example conceptual scenario to present the effectiveness of the XRc in improving the transparency and fairness of the machine learning model to the non-technical stakeholders (board members, executives, and managers). As Figure 3 is shown, the entry point and the exit point are ML Pipeline Phase and Deployment Phase in Figure 2, which makes it our XRc. Such XRc implementation takes over the pre-trained image classification model on Cloud AI Service and uses XAI techniques to explain and analyze the pre-trained model. It then further exposes its representation and API to enable the stakeholders to decide on the deployment phase. At first, the model is identified as an image classification. Then for the XAI techniques implementation, we select the Grad-CAM XAI method paired with an approximation model to obtain the explanations that approximate the cloud AI model. We further analyze the explanation result to gather and filter the information concerning the stakeholders. Finally, we develop the APIs to represent such information and XAI operations, allowing stakeholders to interact efficiently with those processes.

VII. Conclusion

The connection between team-level governance patterns and organization-level patterns can be effectively established using MLOps. The XRc phase provides a clear connection between the two levels of governance, allowing for the navigation and utilization of both patterns. The Design Explanation Microservices and Provenance case study serve as a practical example of this principle-to-practice multi-level governance pattern, showcasing how XAI techniques can be effectively implemented in real-world scenarios to support both team and organizational governance goals. By utilizing the XRc phase in MLOps and examining the case study results, organizations can better understand how to bridge the gap between team-level and organizational-level governance in a way that is both effective and efficient.

Future work in this area could involve exploring new and innovative XAI techniques and methods to enhance further the efficiency and effectiveness of the XRc phase in MLOps. Additionally, continued research into the relationship between team-level and organization-level governance patterns in the context of MLOps will be important in order to fully understand the best practices and strategies for bridging the gap between these two levels of governance.

References

[1] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *CoRR*, vol. abs/1908.09635, 2019. [Online]. Available: http://arxiv.org/abs/1908.09635

[2] Q. Lu, L. Zhu, X. Xu, J. Whittle, D. Douglas, and C. Sanderson, “Software engineering for responsible AI: an empirical study and operationalised patterns,” *CoRR*, vol. abs/2111.09478, 2021.

[3] Q. Lu, L. Zhu, X. Xu, J. Whittle, D. Zowghi, and A. Jacquet, “Responsible ai pattern catalogue: A multivocal literature review,” 2022.

[4] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization,” *CoRR*, vol. abs/1610.02391, 2016. [Online]. Available: http://arxiv.org/abs/1610.02391

[5] F. Hussain, R. Hussain, B. Noye, and S. Sharieh, “Enterprise API security and GDPR compliance: Design and implementation perspective,” *CoRR*, vol. abs/1909.08048, 2019. [Online]. Available: http://arxiv.org/abs/1909.08048

[6] B. Shneiderman, “Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered ai systems,” *ACM Trans. Interact. Intell. Syst.*, vol. 10, no. 4, oct 2020.

[7] ——, “Responsible ai: Bridging from ethics to practice,” *Commun. ACM*, vol. 64, no. 8, p. 32–35, jul 2021.

[8] J. S. Borg, “Four investment areas for ethical ai: Transdisciplinary opportunities to close the publication-to-practice gap,” *Big Data & Society*, vol. 8, no. 2, p. 20539517211040197, 2021.

[9] W. Hussain, M. Shahin, R. Hoda, J. Whittle, H. Perera, A. Nurwidyantoro, R. Shams, and G. Oliver, “How can human values be addressed in agile methods? a case study on safe,” 02 2021.

[10] Q. Lu, L. Zhu, X. Xu, J. Whittle, and Z. Xing, “Towards a roadmap on software engineering for responsible ai,” 03 2022.

[11] T. Shevlane, “Structured access to AI capabilities: an emerging paradigm for safe AI deployment,” *CoRR*, vol. abs/2201.05159, 2022.

[12] A. Jobin, M. Ienca, and E. Vayena, “Artificial intelligence: the global landscape of ethics guidelines,” *CoRR*, vol. abs/1906.11668, 2019. [Online]. Available: http://arxiv.org/abs/1906.11668

[13] Q. Ai and L. N. Ramasamy, “Model-agnostic vs. model-intrinsic interpretability for explainable product search,” *CoRR*, vol. abs/2108.05317, 2021.

[14] P. Ruf, M. Madan, C. Reich, and D. Ould-Abdeslam, "Demystifying mlops and presenting a recipe for the selection of open-source tools," *Applied Sciences*, vol. 11, no. 19, 2021.

[15] S. Shrivastava, D. Patel, N. Zhou, A. Iyengar, and A. Bhamidipaty, "Dqlearn : A toolkit for structured data quality learning," in *2020 IEEE International Conference on Big Data (IEEE BigData 2020), Atlanta, GA, USA, December 10-13, 2020*, X. Wu, C. Jermaine, L. Xiong, X. Hu, O. Kotevska, S. Lu, W. Xu, S. Aluru, C. Zhai, E. Al-Masri, Z. Chen, and J. Saltz, Eds. IEEE, 2020, pp. 1644–1653.

[16] E. Raj, *Engineering MLOps: Rapidly build, test, and manage production-ready machine learning life cycles at scale*. Packt Publishing, 2021.

[17] I. Karamitsos, S. Albarhami, and C. Apostolopoulos, "Applying devops practices of continuous automation for machine learning," *Information*, vol. 11, no. 7, 2020.

[18] S. Mäkinen, H. Skogström, E. Laaksonen, and T. Mikkonen, "Who needs mlops: What data scientists seek to accomplish and how can mlops help?" 2021.

[19] "Mlops workload orchestrator," accessed on August 12, 2022.

[20] "Aws mlops framework," accessed on August 13, 2022.

[21] S. Sharma, *The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise*. Wiley, 2017.

[22] V. Gudivada, A. Apon, and J. Ding, "Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations," *International Journal on Advances in Software*, vol. 10, pp. 1–20, 07 2017.

[23] P. Mattson, C. Cheng, C. Coleman, G. Diamos, P. Micikevicius, D. A. Patterson, H. Tang, G. Wei, P. Bailis, V. Bittorf, D. Brooks, D. Chen, D. Dutta, U. Gupta, K. M. Hazelwood, A. Hock, X. Huang, B. Jia, D. Kang, D. Kanter, N. Kumar, J. Liao, G. Ma, D. Narayanan, T. Oguntebi, G. Pekhimenko, L. Pentecost, V. J. Reddi, T. Robie, T. S. John, C. Wu, L. Xu, C. Young, and M. Zaharia, "Mlperf training benchmark," *CoRR*, vol. abs/1910.01500, 2019. [Online]. Available: http://arxiv.org/abs/1910.01500

[24] G. Fursin, H. Guillou, and N. Essayan, "Codereef: an open platform for portable mlops, reusable automation actions and reproducible benchmarking," *CoRR*, vol. abs/2001.07935, 2020.

[25] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.

[26] J. Zhao, M. Mathieu, R. Goroshin, and Y. LeCun, "Stacked what-where auto-encoders," 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[28] E. R. Sparks, A. Talwalkar, V. Smith, J. Kottalam, X. Pan, J. E. Gonzalez, M. J. Franklin, M. I. Jordan, and T. Kraska, "MLI: an API for distributed machine learning," *CoRR*, vol. abs/1310.5426, 2013.

[29] M. A. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, J. M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Fast and interactive analytics over hadoop data with spark," *login Usenix Mag.*, vol. 37, 2012.

[30] I. Taleb, M. A. Serhani, and R. Dssouli, "Big data quality: A survey," in *2018 IEEE International Congress on Big Data (BigData Congress)*. IEEE, 2018, pp. 166–173.

[31] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2438–2455, 2021.

[32] L. N. Tidjon and F. Khomh, "Threat assessment in machine learning based systems," 2022.

[33] P. Fukas, J. Rebstadt, F. Remark, and O. Thomas, "Developing an artificial intelligence maturity model for auditing," 06 2021.

[34] R. Sadiq, N. Mohd Satar, A. H. Abd Rahman, and S. Goudarzi, "Artificial intelligence maturity model: A systematic literature review," *PeerJ Computer Science*, vol. 7, 08 2021.

[35] B. Fish and L. Stark, "Reflexive design for fairness and other human values in formal models," 07 2021, pp. 89–99.

[36] D. Kißkalt, A. Mayr, B. Lutz, A. Rögele, and J. Franke, "Streamlining the development of data-driven industrial applications by automated machine learning," *Procedia CIRP* vol. 93, pp. 401–406, 2020.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[38] I. Naja, M. Markovic, P. Edwards, and C. Cottrill, "Correction to: A semantic framework to support ai system accountability and audit," in *The Semantic Web*, R. Verborgh, K. Hose, H. Paulheim, P.-A. Champin, M. Maleshkova, O. Corcho, P. Ristoski, and M. Alam, Eds. Cham: Springer International Publishing, 2021, pp. C1–C1.

[39] S. Jentzsch, S. Höhn, and N. Hochgeschwender, *Conversational Interfaces for Explainable AI: A Human-Centred Approach*, 09 2019, pp. 77–92.

[40] A. Perera, A. Aleti, C. Tantithamthavorn, J. Jiarpakdee, B. Turhan, L. Kuhn, and K. Walker, "Search-based fairness testing for regression-based machine learning systems," *Empirical Software Engineering*, vol. 27, 05 2022.

[41] I. Guzey, O. Ucar, N. A. Ciftdemir, and B. Acunas, "Context-dependent explainability and contestability for trustworthy medical artificial intelligence: Misclassification identification of morbidity recognition models in preterm infants," 2022.

[42] M. Treveil, N. Omont, C. Stenac, K. Lefevre, D. Phan, J. Zentici, A. Lavoillotte, M. Miyazaki, and L. Heidmann, *Introducing MLOps*. O'Reilly Media, 2020.

[43] H. Mehmood, P. Kostakos, M. Cortes, T. Anagnostopoulos, S. Pirttikangas, and E. Gilman, "Concept drift adaptation techniques in distributed environment for real-world data streams," *Smart Cities*, vol. 4, no. 1, pp. 349–371, 2021.

[44] F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., *Automatic Machine Learning: Methods, Systems, Challenges*. Springer, 2019.

[45] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Auto-sklearn: Efficient and robust automated machine learning," F. Hutter, L. Kotthoff, and J. Vanschoren, Eds. Springer, 2019, pp. 123–143.

[46] N. Muralidhar, S. Muthiah, P. Butler, M. Jain, Y. Yu, K. Burne, W. Li, D. Jones, P. Arunachalam, H. S. McCormick *et al.*, "Using antipatterns to avoid mlops mistakes," *arXiv preprint arXiv:2107.00079*, 2021.

[47] T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.

[48] T. Granlund, A. Kopponen, V. Stirbu, L. Myllyaho, and T. Mikkonen, "Mlops challenges in multi-organization setup: Experiences from two real-world cases," in *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*. IEEE, 2021, pp. 82–88.

[49] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[51] D. A. Tamburri, "Sustainable mlops: Trends and challenges," in *2020 22nd international symposium on symbolic and numeric algorithms for scientific computing (SYNASC)*. IEEE, 2020, pp. 17–23.

[52] B. Mons, *Data stewardship for open science: Implementing FAIR principles*. Chapman and Hall/CRC, 2018.

[53] A. Maydanchik, *Data quality assessment*. Technics publications, 2007.

[54] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 1135–1144.

[55] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017. [Online]. Available: https://arxiv.org/abs/1705.07874