

The MLOps Workbook

A Guided Online Course for Getting Started with MLOps

December 5, 2023

Eine gemeinsame
Initiative

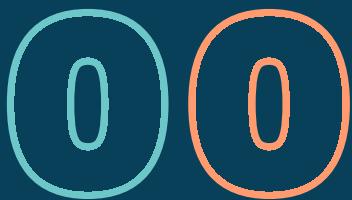
UNTER
NEHMER
TUM



Ipai



This course, which is available under the Creative Commons Attribution 4.0 International License (CC BY 4.0, <https://creativecommons.org/licenses/by/4.0/>), may be freely shared, adapted, and used for commercial purposes provided it is properly credited to "AppliedAI Institute for Europe gGmbH".



Introduction

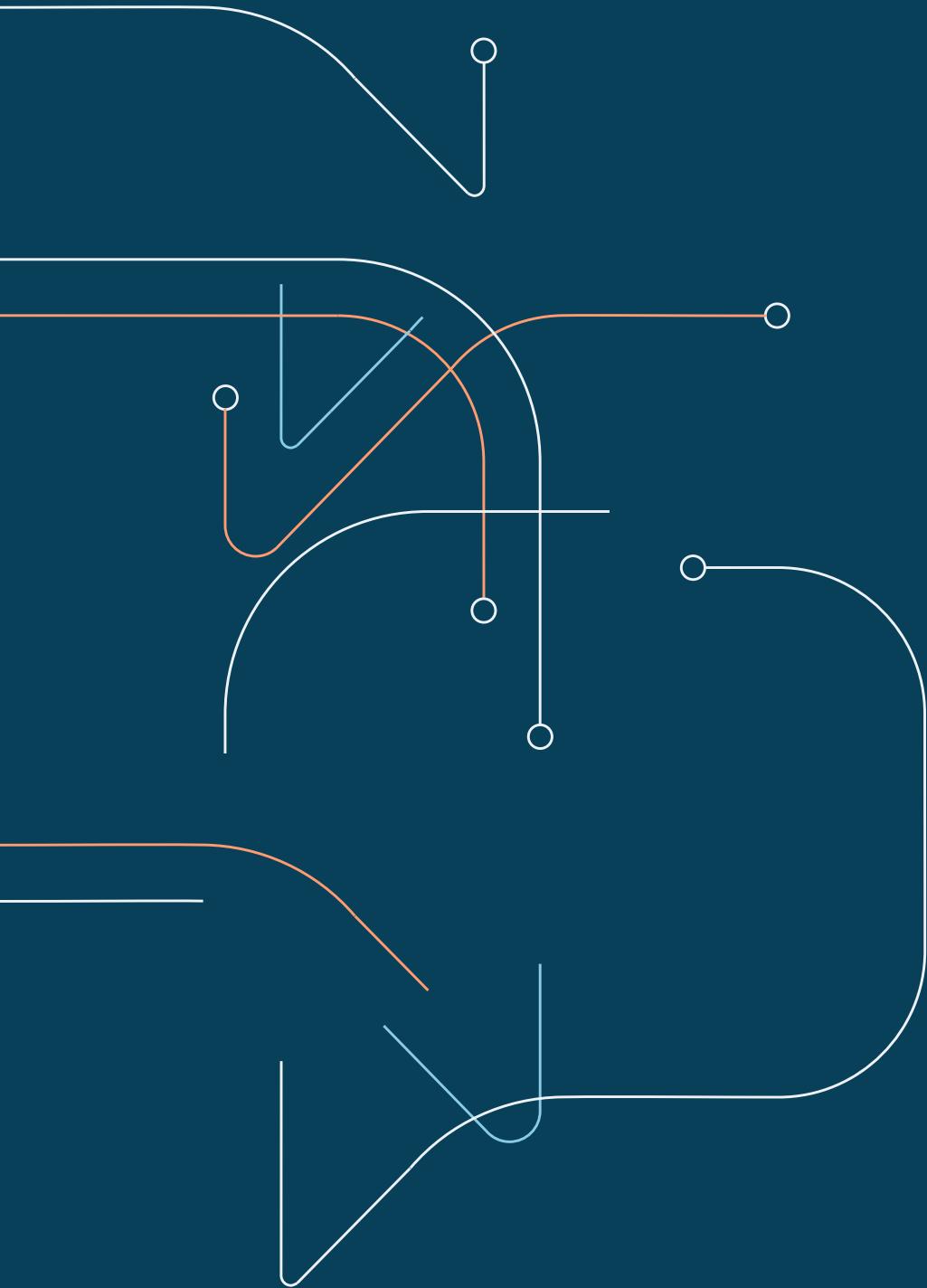
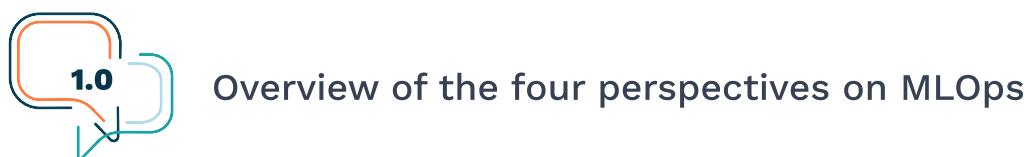


Table of contents

The ML Workbook is organized into seven modules that follow the ML Lifecycle, a key perspective that we will introduce in this course. Note that this workbook does not use page numbers, but IDs. These IDs are located in the top corner of each page.

**1.0**

Overview of the four perspectives on MLOps

**2.0**

The Scoping Stage of the ML Lifecycle

**3.0**

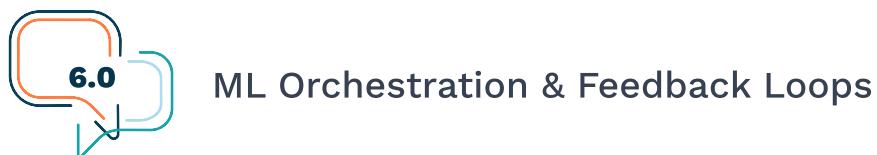
The Data Engineering Stage of the ML Lifecycle

**4.0**

The Modeling Stage of the ML Lifecycle

**5.0**

The Deployment Stage of the ML Lifecycle

**6.0**

ML Orchestration & Feedback Loops

**7.0**

Credits

What you will learn from this book

This workbook is designed to help you and your team understand the process of professionalizing your machine learning with MLOps. By the time you complete this workbook, you will have acquired a solid grasp of the essential concepts required to deliver value using MLOps.

After completing the workbook and closely following along with the materials, you will be able to:

-  Describe the components and functions of the four perspectives ML Principles, ML Lifecycle, ML Accountabilities, and the appliedAI Project Phases Framework.
-  Explain the organizational factors of MLOps, which involves outlining the involvement of the ML Accountabilities throughout the ML Lifecycle and ML Project Phases Framework.
-  Describe ML workflow improvements that enhance the maturity of your MLOps processes. The aim is to begin a discussion on scaling the number of ML projects and reducing time-to-market for ML models.
-  Realize how to put the content of the course into practice through a best-practice prototype of an ML system.

Is this course for me?

If you are still unsure whether you should spend your time on this course, you can do a self-assessment here. Tick each statement that applies to you or your team. The more you tick, the more likely you are to benefit from this course.

Few of our ML systems make it into production.

I do not know where I can find the data in my company to generate insights and apply ML.

I have a rough idea of what MLOps is, but I don't grasp it fully.

I am working in an ML team or plan to work in an ML team.

Our team does a lot of manual work to bring ML models into production without having a standardized process.

We have a use case where governance and compliance is required.

We don't have a well-defined language for talking about ML projects between technical and non-technical team members.

We don't have a clear roadmap or guidelines on what factors to consider when taking an AI project from the idea stage to putting it into production.

Learning with the video course

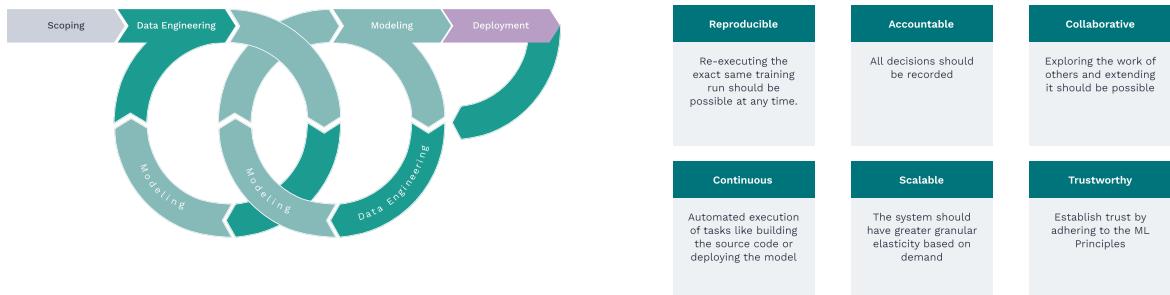
The ML Workbook comes with a customized video course. In the video course, our instructors explain the content of the workbook in detail and provide solutions to the exercises. For an optimal learning experience, we recommend that you use the ML Workbook together with the video course.



INTRODUCTION

The four perspectives on MLOps

Throughout the course, we will explore MLOps from four perspectives, which focus on different aspects of developing ML systems. We will revisit them and draw connections between them.



The ML Lifecycle

The ML Lifecycle is a meta-model that outlines the stages through which modern machine learning systems progress.

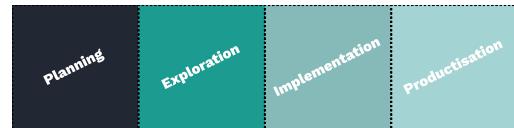


The ML Accountabilities

The accountabilities describe the various responsibilities involved in developing an ML system.

The MLOps Principles

The six MLOps Principles serve as the foundation upon which the ML Lifecycle is realized.



The appliedAI

Project Phases Framework

The framework outlines four phases for implementing an ML project.

Resources

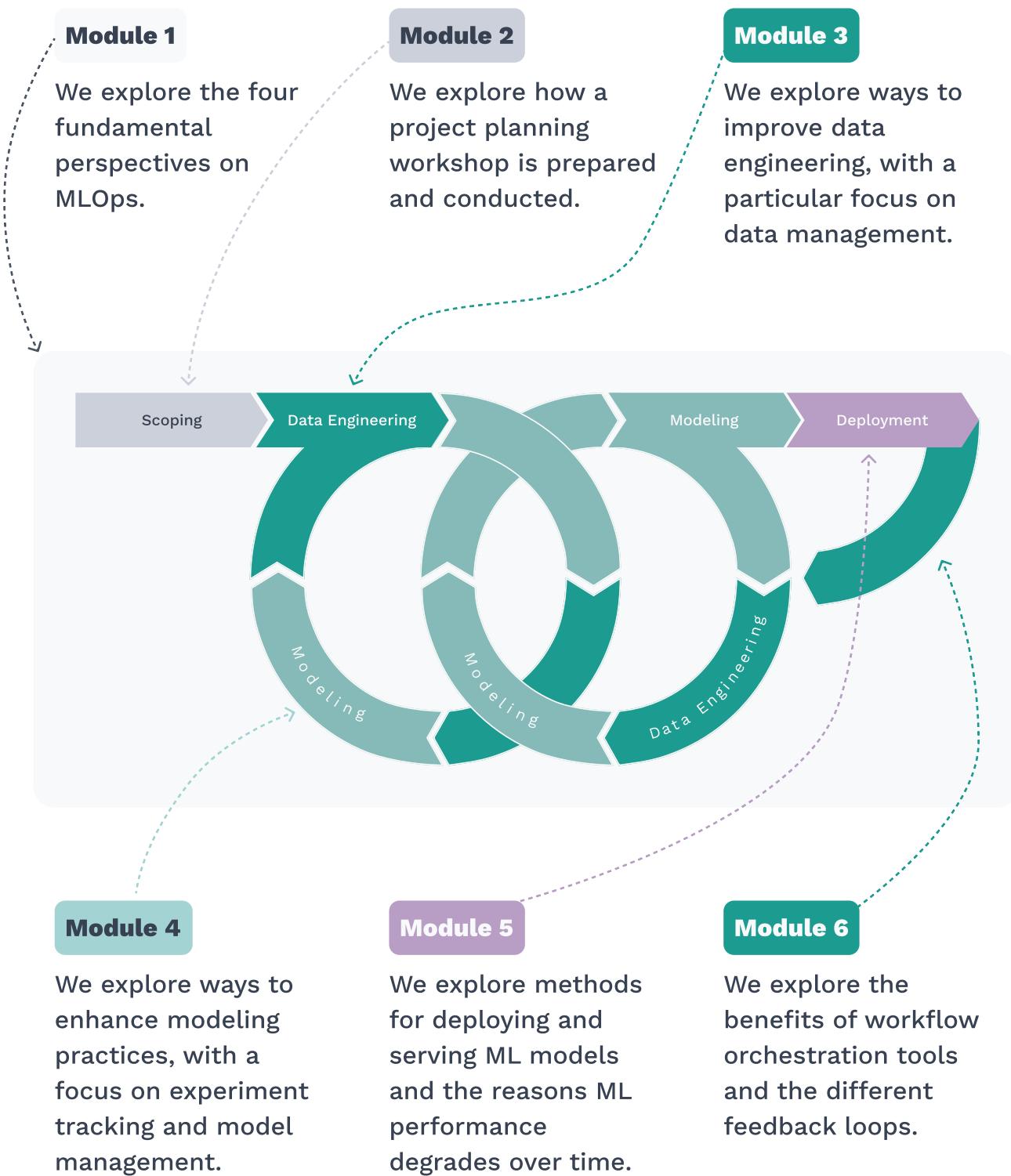
Our instructor Alexander Machado, in collaboration with AI industry experts, has developed some of these MLOps perspectives over the past few years, enabling enterprises to efficiently bring AI projects to production.

Find out more:

- ML Lifecycle and AI Project Phases: Machado, Waldmann: “The Enterprise Guide to ML”
- ML Accountabilities: Machado, Mynter: “ML Skill Profiles: An Organizational Blueprint for Scaling Enterprise ML”

Module overview

We have structured this course around the ML Lifecycle, which describes the stages through which modern ML systems progress. The first module offers an introduction to the ML Lifecycle and the other three perspectives on MLOps. The later modules delve deeper into the other stages of the ML Lifecycle.



Module structure

Learning is not a spectator sport, which is why we have crafted this workbook with a strong focus on active participation. Our aim is to familiarize you with the fundamental perspectives on MLOps. All six modules follow a consistent structure to help you achieve the defined learning objectives effectively.



Intended learning objectives

At the beginning of each module, you will find a set of learning objectives that we want you to accomplish.



Main content in small segments

The main content is presented in digestible pieces. On about half of the pages, you are asked to actively participate and express your thoughts and understanding of specific parts of the material.



Practical recommendations

After covering the content, we provide practical recommendations for you and your team to implement the ideas.

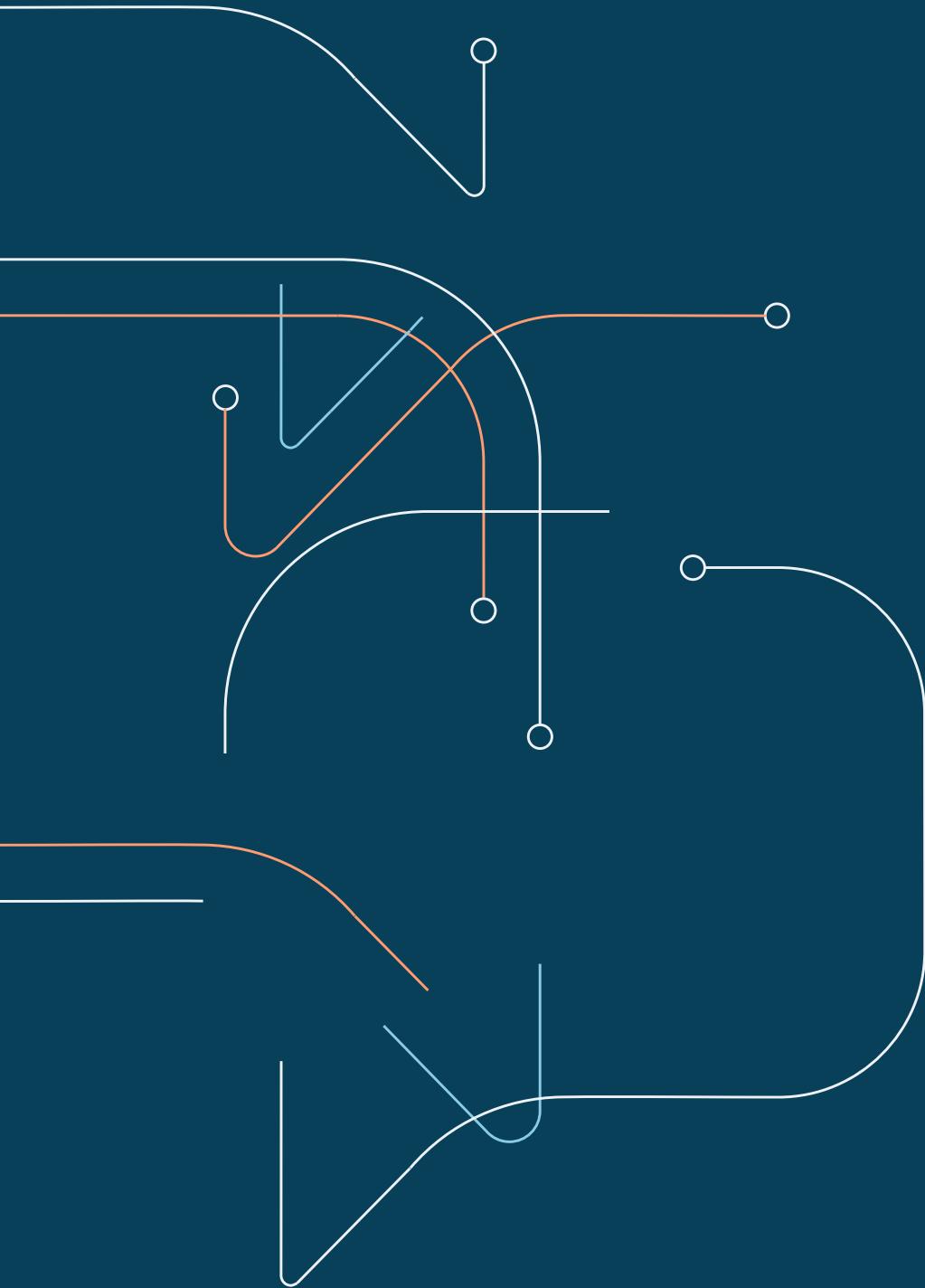


Self-assessment of comprehension

As the module draws to a close, you will find a self-assessment that serves as a tool to assess your comprehension of the module's content.

01

Overview



Intended learning objectives

In this module, we delve into the fundamental perspectives on MLOps. As we progress, we continuously revisit these perspectives, striving to uncover their interconnections. By exploring the four perspectives, we understand MLOps through diverse lenses, each significant in tackling the complexities of bringing ML systems into production.

By the end of this module, you will have developed the following proficiencies:

- ✓ Explain why the existence of hidden technical debt calls for MLOps processes.
- ✓ Describe the four perspectives on MLOps: ML Lifecycle, ML Accountabilities, ML Principles and the appliedAI Project Phases Framework.
- ✓ List and describe the stages and phases of the ML Lifecycle
- ✓ Distinguish the focus of the Stage View of the ML Lifecycle from the Phase View of the ML Lifecycle.
- ✓ Explain the function of the phases of the appliedAI Project Phases Framework.
- ✓ Describe the importance of the ML Accountabilities during the ML Lifecycle and the appliedAI Project Phases Framework.
- ✓ Describe how ML projects are brought into production from the lens of the four perspectives.

Before you begin, we recommend that you take a look at the contents of the module. Take a few minutes to go through each page and look at the headings and visuals. Try to get an overview of the module's content. When you are done, go to the last page of the module and write down a few questions you would like to have answered at the end of the module. We will ask you to review these questions later after you have worked through the content.

Can you answer these questions?

Mature MLOps practices are obvious when ML teams can quickly answer questions about the status and functionality of their ML systems. Below are some questions that a team with a mature system can answer quickly. Try to determine if you could answer the questions within a reasonable amount of time.

What was the accuracy of the June 12, 2022 model on the training and test sets, and were there any issues with overfitting or underfitting?



What were the data sources and preprocessing steps used to train the models for a specific use case?



What version of the code base was used to train the August 4, 2023 model, and has the code changed since then?



What was the full list of models trained on the April 26, 2023 dataset, and what were the differences between them?



How many models were deployed in the last three months?



What was the accuracy of the last 40 models created?



How do you protect your team from accidentally changing training data files?



Has the input data for your model changed recently (e.g. new scheme, new distribution)?



The perils of Hidden Technical Debt

If you couldn't answer some of the questions on the previous page, your ML system may have a problem called hidden technical debt. It refers to the unnoticed and accrued maintenance costs resulting from rapid development. An ML system in production should have little hidden technical debt, but in development it's up to your team to decide how much cost is tolerable.

"As the machine learning (ML) community continues to accumulate years of experience with live systems, a wide-spread and uncomfortable trend has emerged: developing and deploying ML systems is relatively fast and cheap, but maintaining them over time is difficult and expensive."

1

Because ML teams often develop ML systems fast, the systems are hard to maintain

"As with fiscal debt, there are often sound strategic reasons to take on technical debt. Not all debt is bad, but all debt needs to be serviced."

2

Sometimes it makes sense to accumulate technical debt to get a proof of concept quickly, but the team needs to service it later in the project

"Deferring such payments results in compounding costs. Hidden debt is dangerous because it compounds silently."

3

MLOps tries to avoid building technical debt by automating processes and increasing abstraction in ML software development

Resources

<https://proceedings.neurips.cc/paper/2015/file/86df7dcfd896fcf2674f757a2463eba-Paper.pdf>

What's your Hidden (Technical) Debt?

Hidden (Technical) Debt can come in many forms. Examine the various technical debts in the following workflow and assess whether they have impacted your system.

We didn't keep a continuous record of the results of our experiments.

We could not automatically detect changes in the schema of external data.

Hyperparameter tuning parameters were hard coded and not stored in configuration files.

We neglected to evaluate whether a new model outperforms the old model when we update it.

We were unable to determine if the expected value of our data was outside the expected range.

We did not use Git branches to test certain data preparation experiments.

We stored our data on a local computer or on a hard drive instead of a shared infrastructure.

Switching between models lead to temporary interruptions in operation.

We didn't automatically re-train our models when their performance dropped.

When planning the project we didn't look into the existing data.

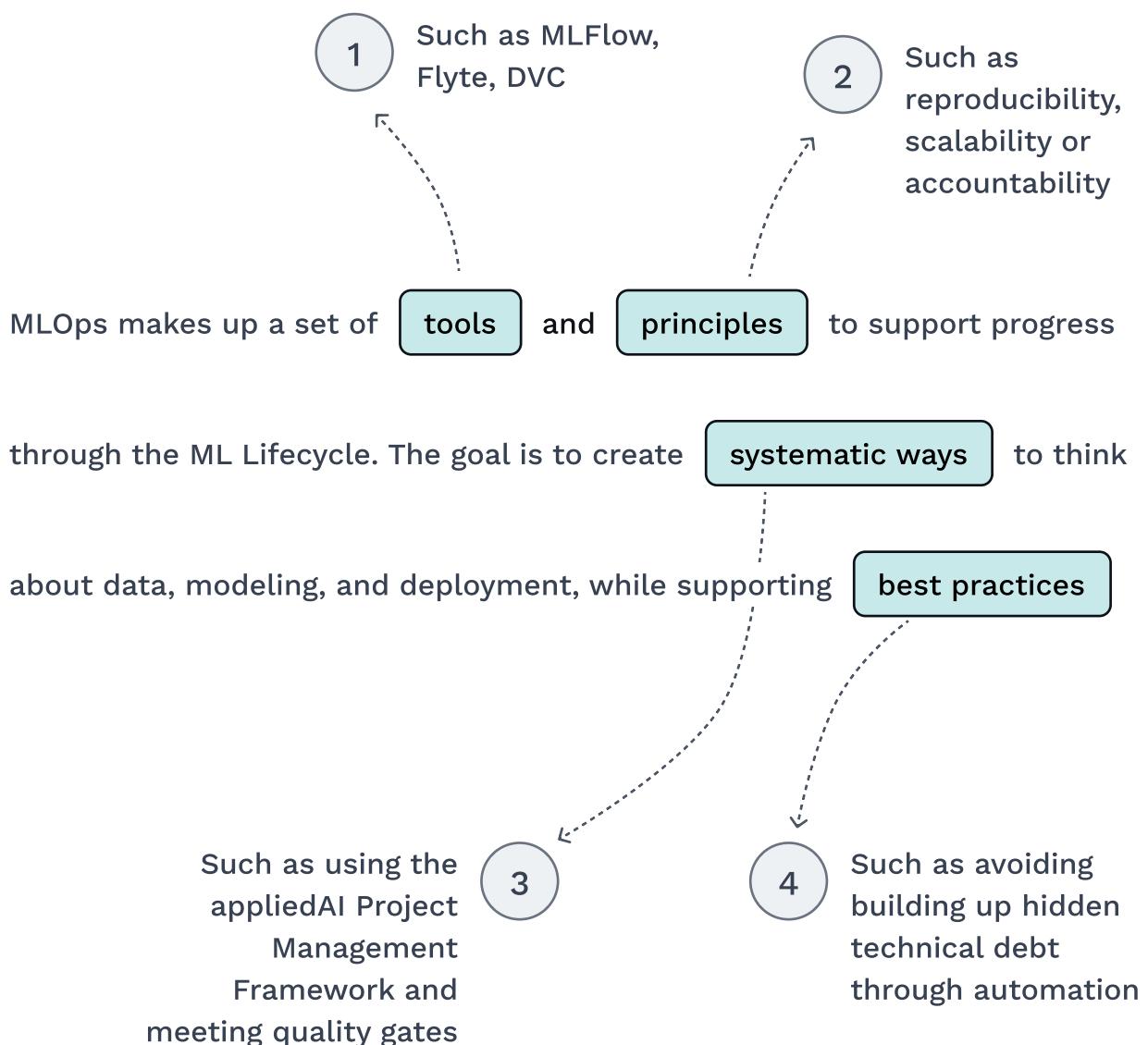
We budgeted the ML project only until the deployment of the model.

We determined the team size and the working time spontaneously.

We didn't prioritize use cases at the beginning of the project.

The definition of MLOps

MLOps is designed to avoid Hidden Technical Debt by creating systematic ways to bring ML systems into production. Below is a definition of MLOps that we will use in this course.

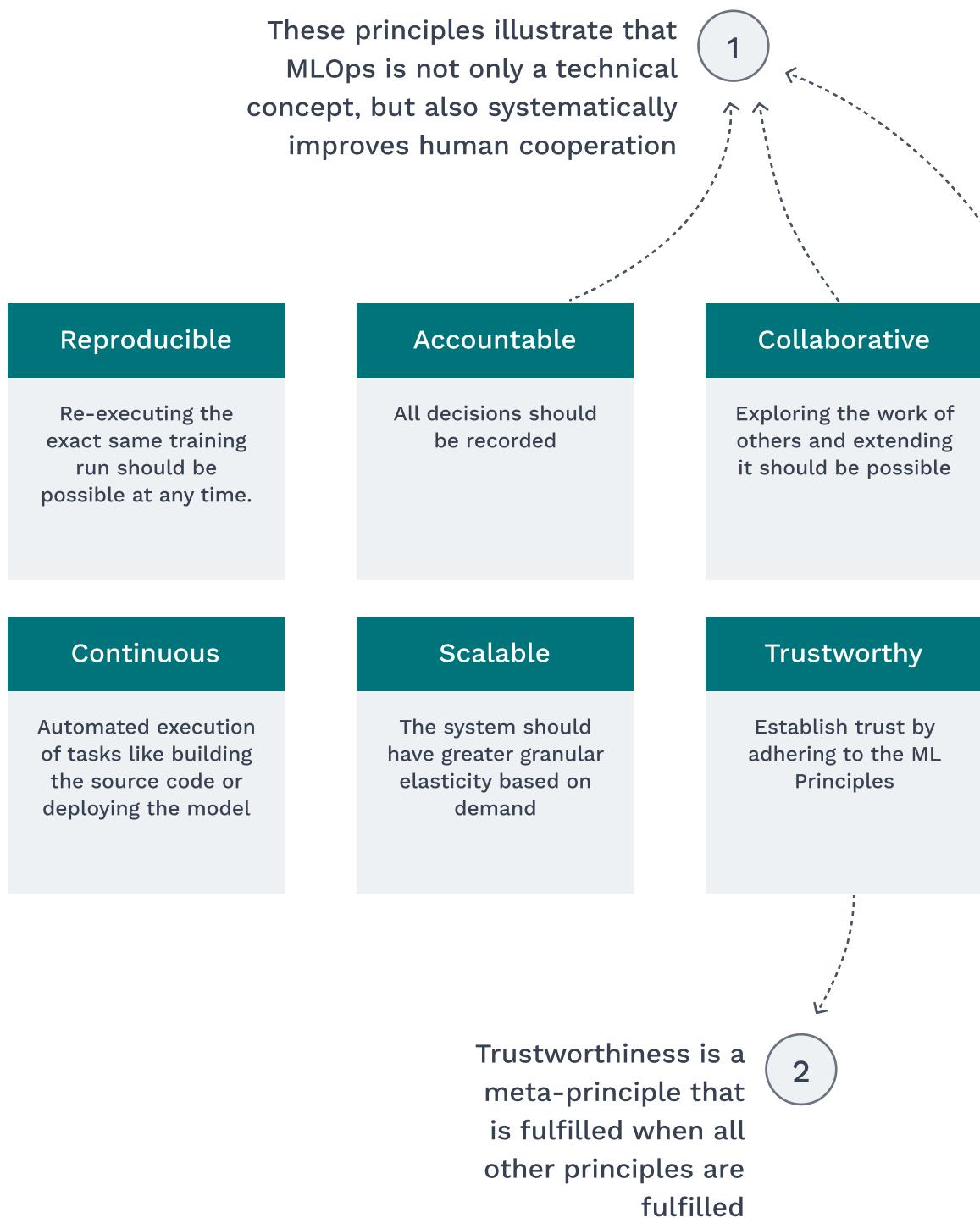


Resources

<https://appliedaiinitiative.notion.site/The-enterprise-guide-to-ML-a389dbdf244143f7a690b4e1980444f4>

The MLOps Principles

On the next pages we will introduce the four perspectives of MLOps. The key perspective is the ML Principles, which are the foundation for every other perspective. If you have doubts about how your team is developing your ML system, come back to the ML Principles and consider whether you have put them into practice.



Resources

<https://ml-ops.org/content/mlops-principles>

Putting the MLOps Principles into practice

Below you will find the six MLOps principles. For each principle, try to match the statement that best corresponds to it.

1

“We are able to access previous version of the code and datasets.”

2

“We can trigger a model retrain within a few minutes when new data comes in.”

3

“The data catalog shows who is the responsible person (data owner) for a given dataset.”

4

“Model training performance can be checked by everyone on the team in the experiment tracking dashboard.”

5

“Our model does not discriminate against a minority class.”

6

“Our model can be automatically deployed on more servers when the load on the API increases.”



Reproducible



Accountable



Collaborative



Continuous



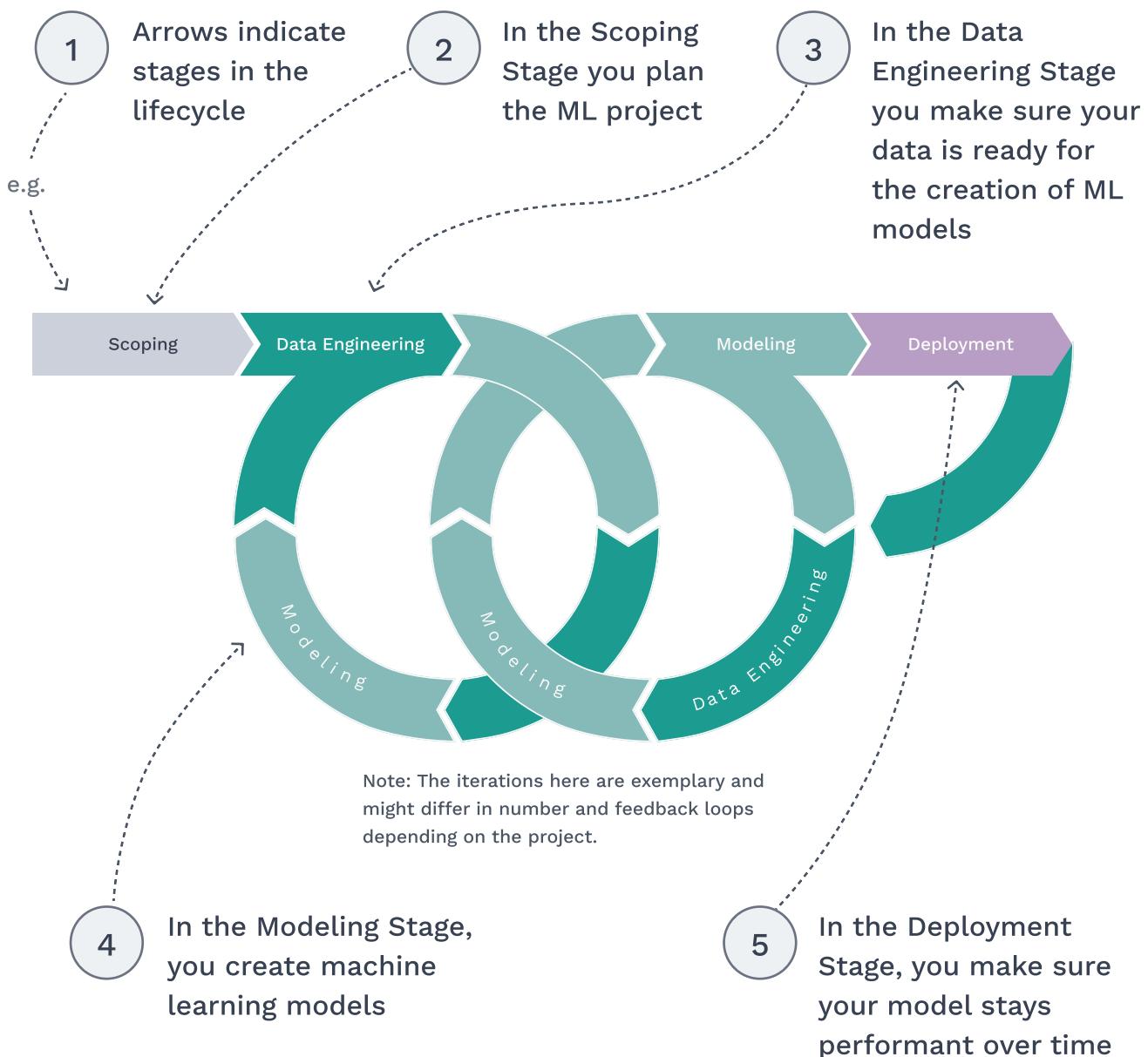
Scalable



Trustworthy

The Stage-View of the ML Lifecycle

The ML Lifecycle is a perspective that describes the cyclical process of developing, training, and deploying machine learning models with large amounts of data. Every machine learning system follows the stages outlined in the lifecycle to a certain extent.

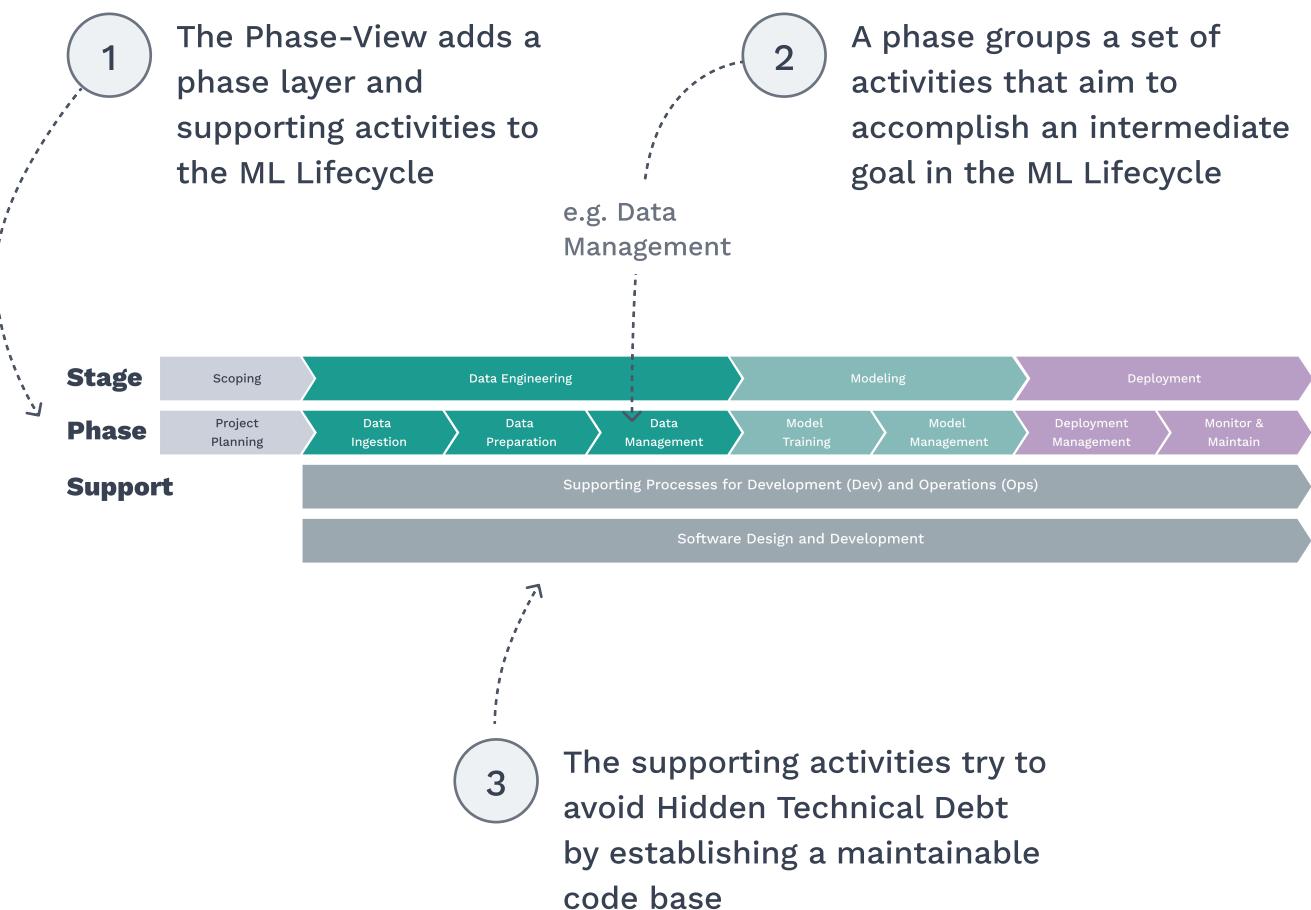


Resources

<https://www.appliedai.de/en/hub-en/enterprise-guide-to-machine-learning>

The Phase-View of the ML Lifecycle

We can also unfold the ML Lifecycle to reveal the underlying phases of each stage. Also, this view shows the supporting software engineering processes from the Data Engineering to the Deployment Stage.

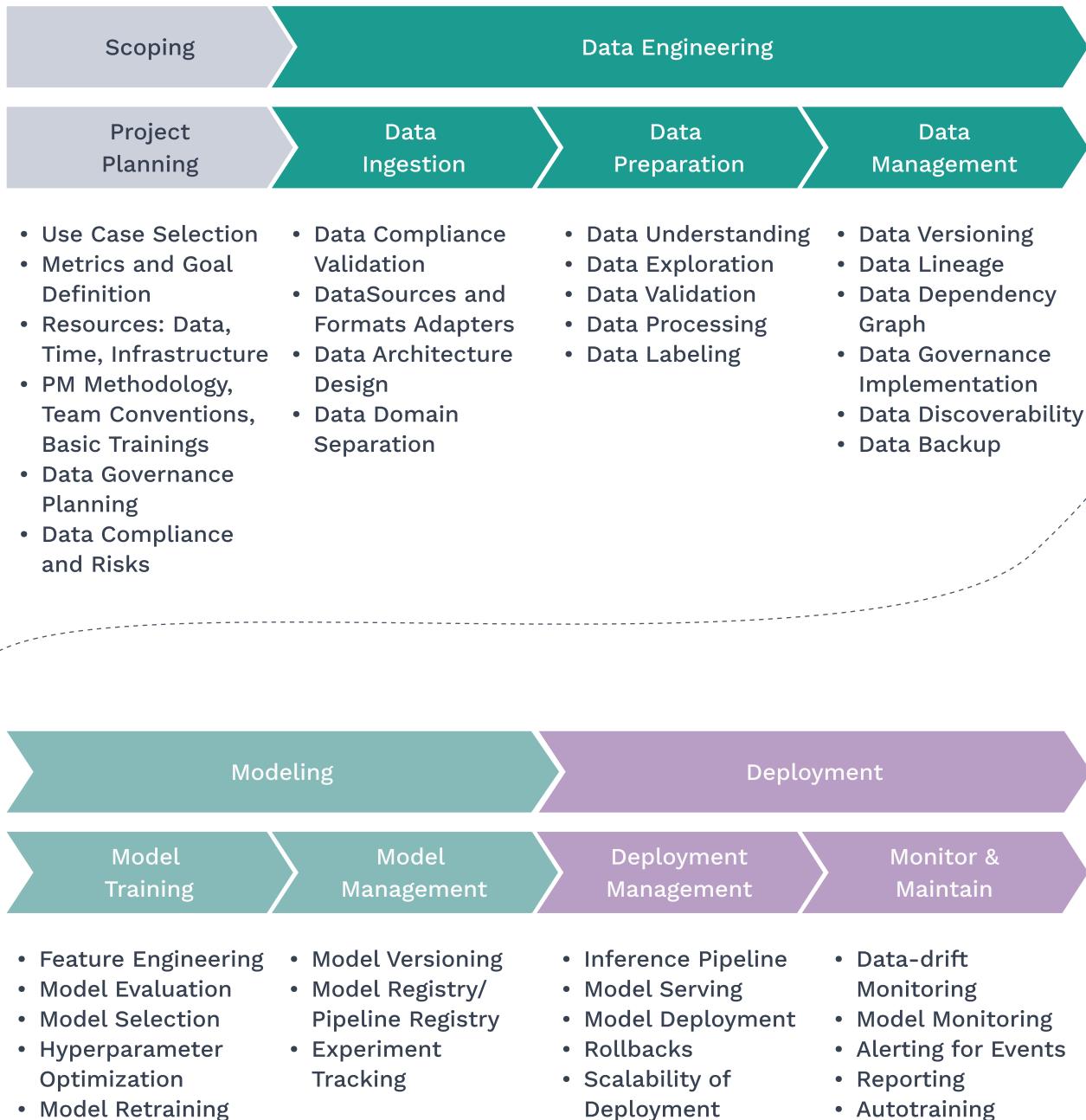


Resources

<https://www.appliedai.de/en/hub-en/enterprise-guide-to-machine-learning>

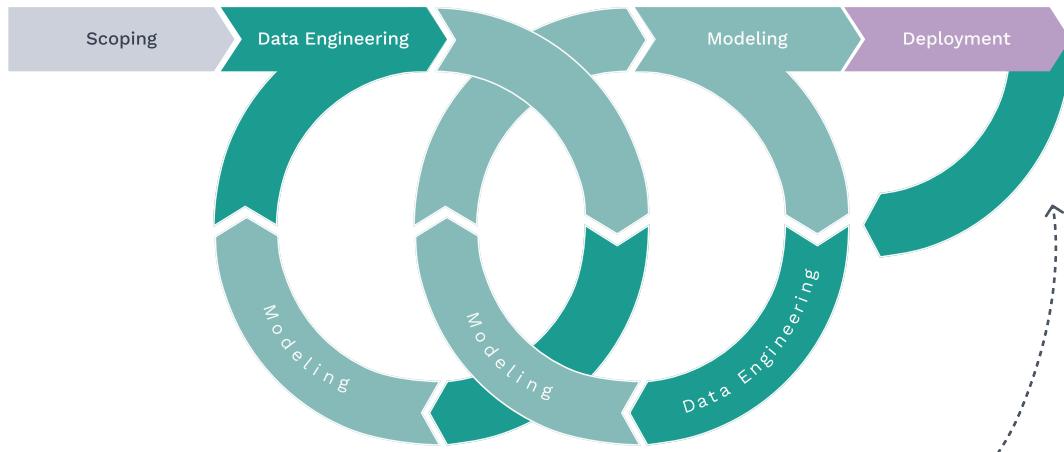
Activities of the ML Lifecycle

The Phase-View of the ML Lifecycle has another layer that we have not yet mentioned, the Activity layer. This layer describes phase-specific activities that ML teams can carry out. Remember that no ML team performs all of these activities, but only a subset of them.

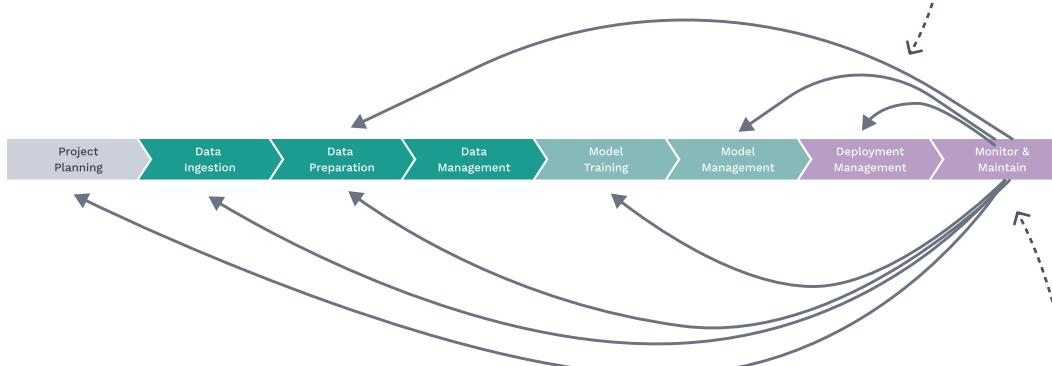


Feedback loops

Feedback loops play a pivotal role in the ML Lifecycle, as they enable information to circulate back to earlier phases. Such feedback loops exist in both the Stage-View and the Phase-View ML Lifecycles.



A feedback loop refers to an action that is activated from one stage of the lifecycle, then influencing and triggering an action in one of the preceding phases



While the Deployment Stage serves as the starting point for all feedback loops in this diagram, it's important to note that these loops can originate from any phase within the lifecycle

2

ML Accountabilities throughout the ML Lifecycle

Although machine learning teams may vary in size and composition, they all share the same accountabilities. An accountability is a skill profile. One person can take on multiple accountabilities. Note that these accountabilities are a simplified and high-level overview, and your team may have more or fewer of them. Try to identify which accountabilities you currently undertake or plan to undertake.

Accountability	Description	I am accountable
 Product Owner Paige	<p>They steer the success of a machine learning project and/or product from a business perspective. They work closely with stakeholders to understand business goals and requirements. Also, they collaborate with the development team to make sure that the product meets these goals.</p>	<input type="checkbox"/>
 Data Engineering Damian	<p>They ingest and process large volumes of data and create efficient, scalable pipelines that enable fast and accurate data access and analysis.</p>	<input type="checkbox"/>
 Data Science Doris	<p>Their task is to develop machine learning models that solve business problems and answer ML-related research questions such as feature engineering. To this end, they perform use-case-specific data preparation, data ingestion, and modeling activities.</p>	<input type="checkbox"/>
 Data Steward Deborah	<p>They guarantee the quality and compliance of data. They validate that data policies were followed in each use case and enforce data policies and procedures. They also help to identify data-related risks and opportunities.</p>	<input type="checkbox"/>
 ML Engineering Matthew	<p>They are responsible for designing, implementing, and preparing ML models for deployment. They work closely with data scientists and solution architects to develop and integrate these models into existing systems to guarantee optimal performance.</p>	<input type="checkbox"/>
 Solution Architect Samantha	<p>They are responsible for the design and implementation of complex technical systems. They work closely with stakeholders to identify the best technical architectures and oversee their implementation, debugging, and performance.</p>	<input type="checkbox"/>

In the words of the ML Accountabilities

For this exercise, try to match the following statements to their corresponding ML Accountabilities.

1

"I ensure that my model's performance is optimized to the fullest extent possible."

3

"I ensure that deploying a new model does not cause any disruptions to traffic."

5

"I ensure consistent model outcomes for identical datasets."

2

"I ensure that the data platform is capable of working with a wide range of datasets."

4

"I ensure that project requirements are clearly defined."

6

"I ensure that all data governance guidelines are adhered to."

for example

5



Data Engineering
Damian



ML Engineering
Matthew



Product Owning
Paige



Solution Architect
Samantha



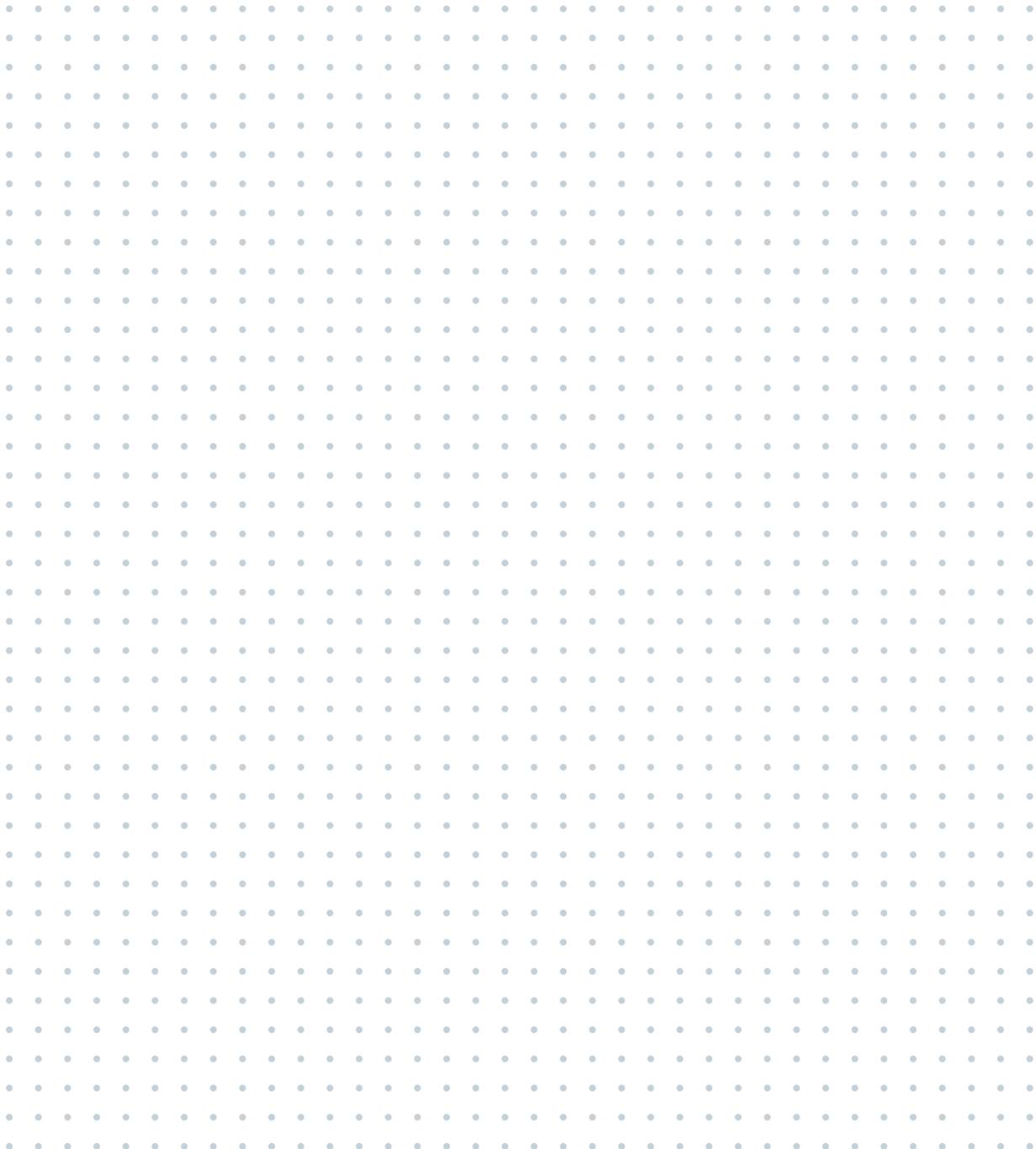
Data Science
Doris



Data Steward
Deborah

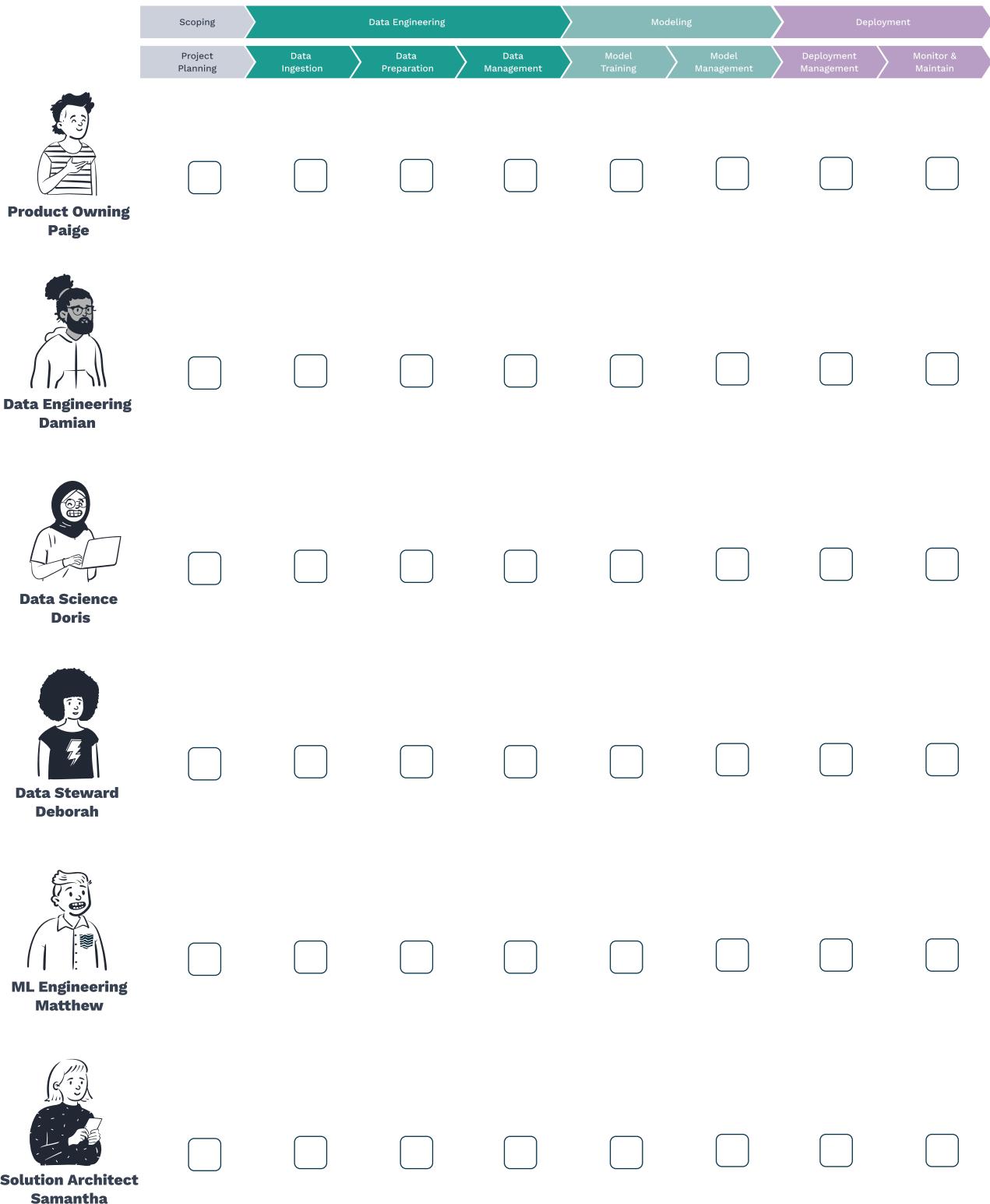
The ML Accountabilities in your ML team

Now that you have identified the accountabilities for which you are accountable, try to outline how the six accountabilities are distributed within your team. Remember that one person can take on multiple accountabilities. Choose the type of visualization that works best for you.



For which phases of the ML Lifecycle are the ML Accountabilities responsible?

The accountabilities place emphasis on different phases of the ML Lifecycle. Try to identify for which phases of the lifecycle the accountabilities are responsible.



ML projects have to handle uncertainty

ML projects are notorious for their uncertainty. Or, as the saying goes, more than 90% of ML systems never make it into production. Uncertainty can be described as a situation where a group of people or an individual has imperfect or unknown information to predict the outcome of a future event. In this exercise, we ask you to do two things. Identify what makes ML systems uncertain and identify strategies to mitigate uncertainty in ML project management.

What makes ML systems uncertain?

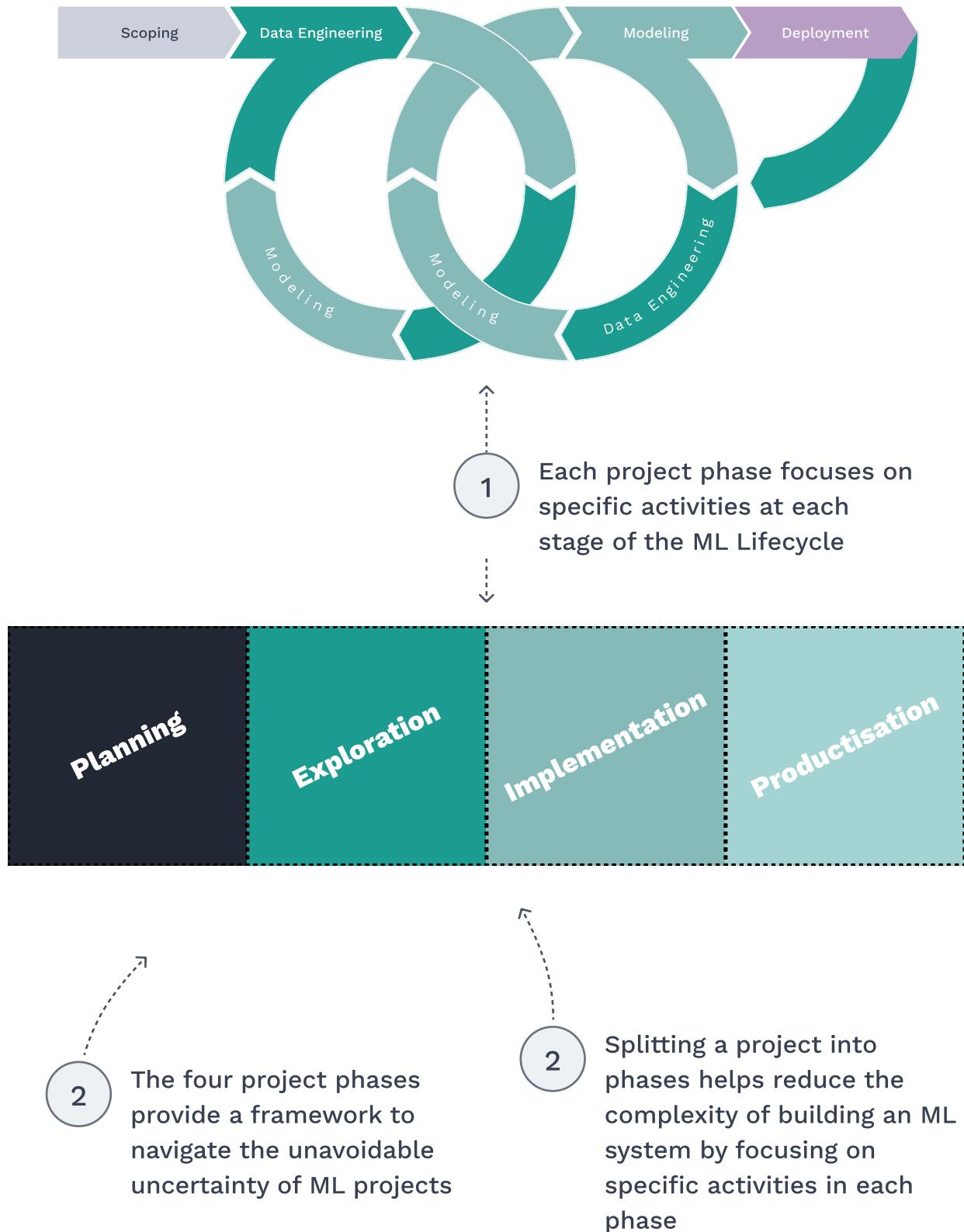
- A uniform grid of small, light gray dots on a white background. The dots are arranged in a regular, repeating pattern across the entire area.

How can uncertainty be mitigated in ML systems?

- A uniform grid of small, light gray dots on a white background. The dots are arranged in a regular, rectangular pattern, covering the entire area.

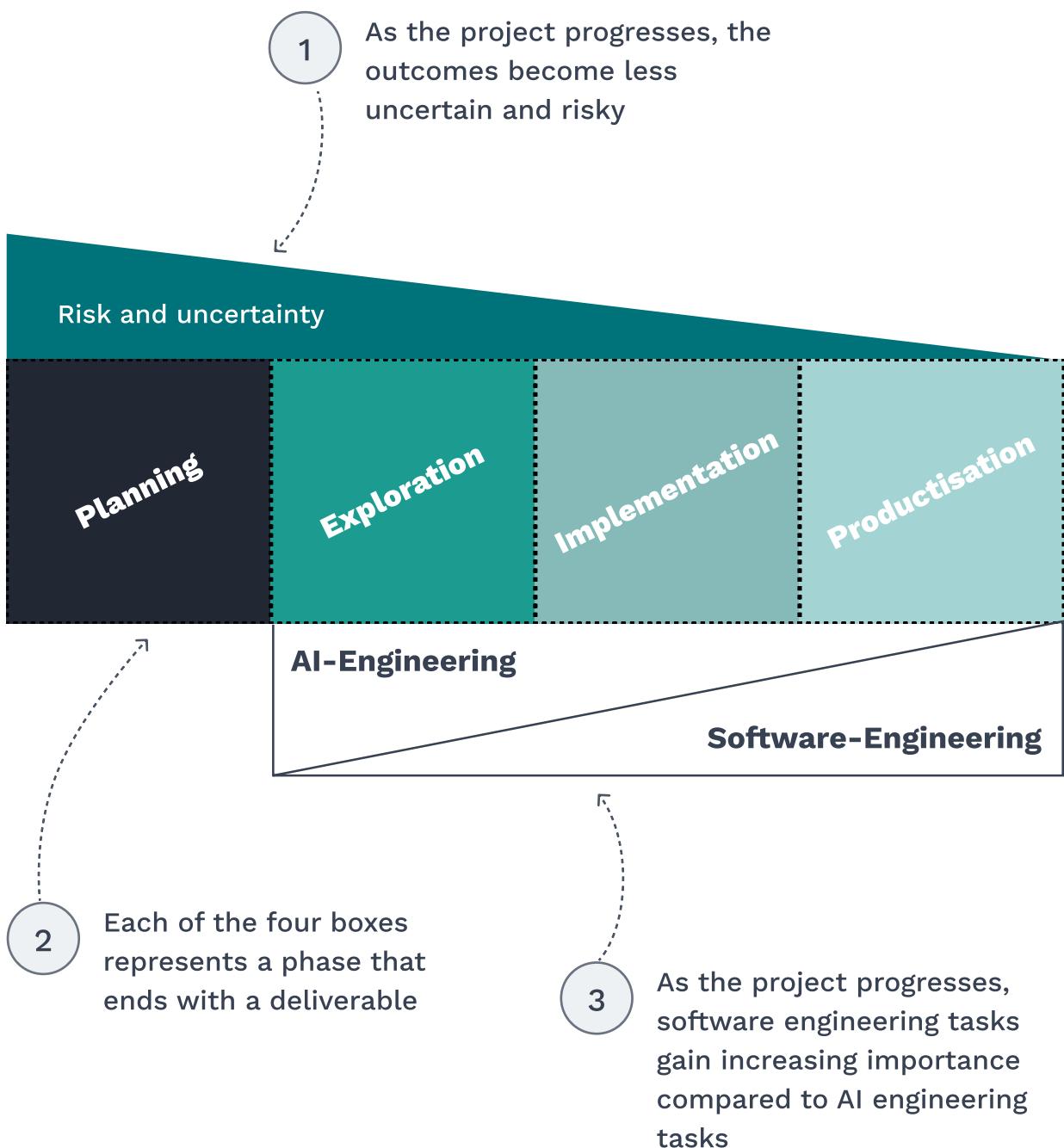
The appliedAI Project Phases Framework

The appliedAI Project Phases Framework was developed to reduce risk in ML system development and to manage the uncertainty associated with it. Different project phases characterize the framework.



An overview of the appliedAI Project Phases Framework

The four project phases differ not only in their core activities but also in the level of risk involved in taking the system to production, as well as the share of software engineering work required.



Unpacking the project phases

Below is a description of each of the four project phases, including their duration, the recommended project management method, and the key deliverables for each phase.

Planning	Exploration	Implementation	Productionisation
In this phase, the team ideates, prioritizes, and selects AI use cases. The team then defines the project for the selected use cases.	In this phase, the team explores the feasibility of the project based on a small dataset with a baseline model.	In this phase, the team researches and explores the most promising models and engineers further features based on the available dataset.	In this phase, the team integrates the model into a production system and automates the ML Lifecycle activities.
Project management			
Workshop-based	e.g. Kanban	e.g. SCRUM	e.g. SCRUM
Deliverable			
Template/Canvas	Feasibility report	Feasibility report	ML system
Relative duration			
A few days	4 to 6 weeks	2 to 6 months	A few months

1

The duration of the individual phases depends on the project. These time frames should be considered as rough suggestions for structuring the relative timing of each phase

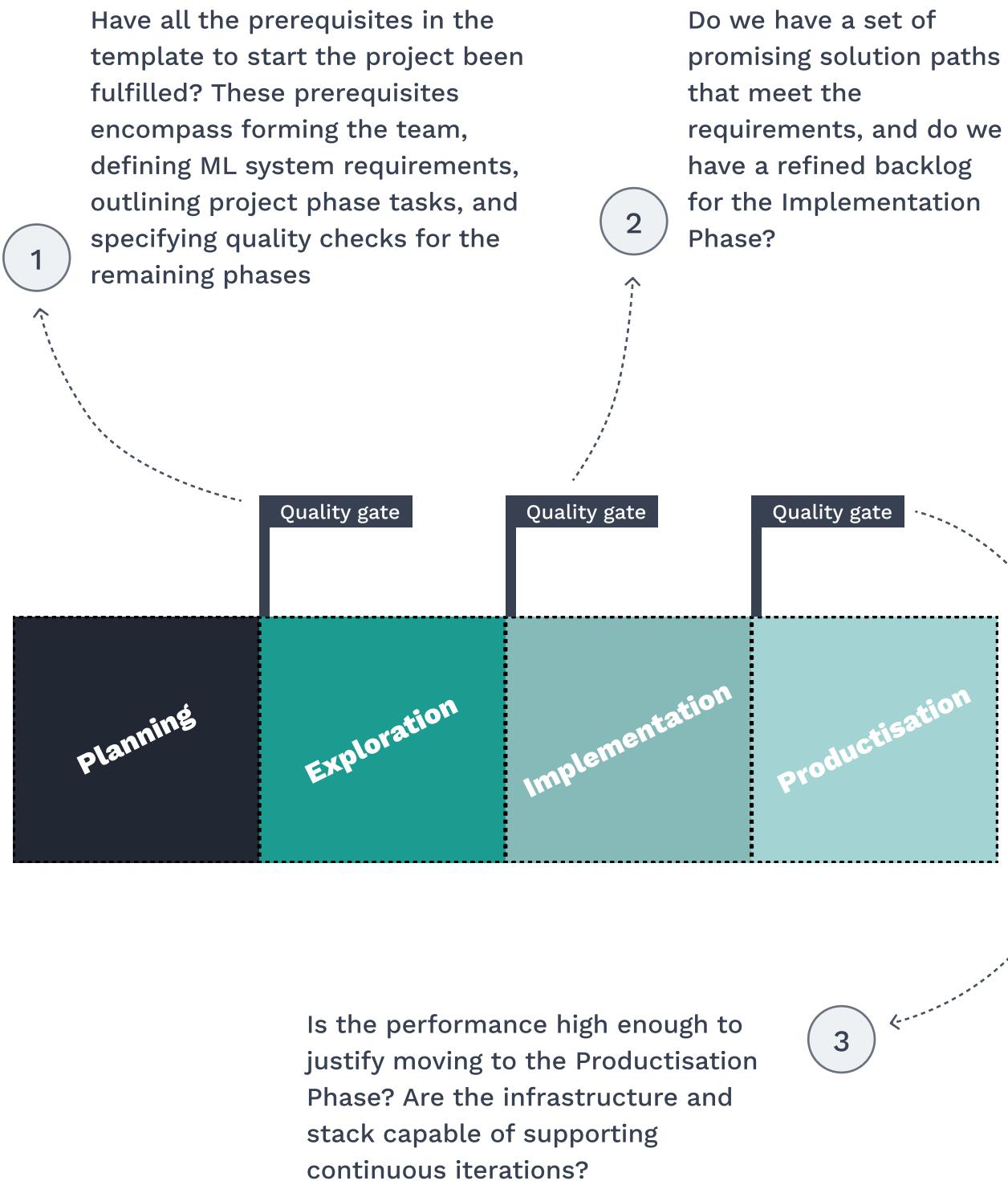
Mapping the ML Accountabilities to the project phases

Each accountability carries varying significance in each project phase. In this exercise, determine which accountability becomes most critical in each phase. Additionally, reflect on your role during each project phase.

	Planning	Exploration	Implementation	Productisation
Product Owning Paige	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Engineering Damian	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Science Doris	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Steward Deborah	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ML Engineering Matthew	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Solution Architect Samantha	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

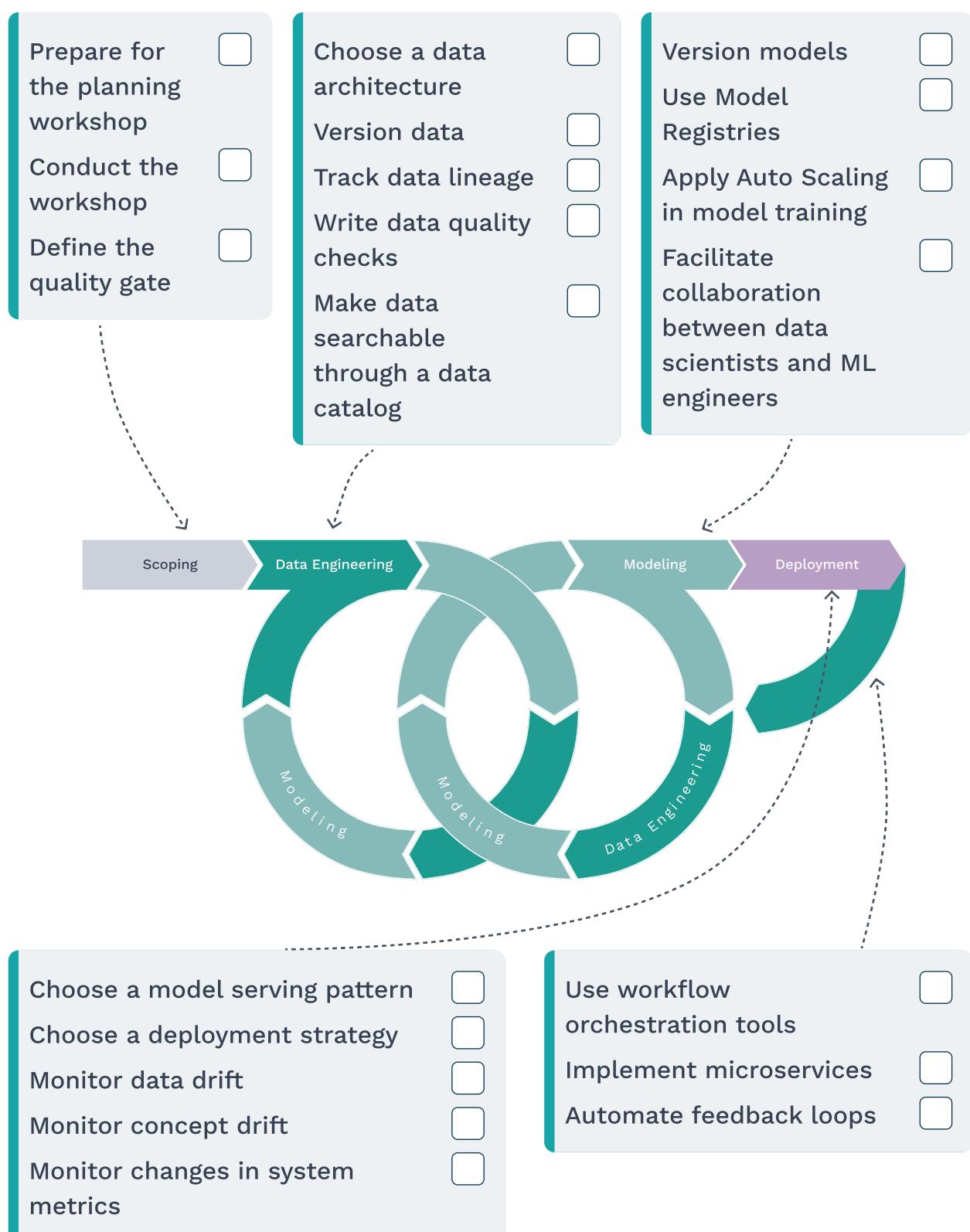
Quality gates

A quality gate serves as a midway assessment to determine whether it is justifiable to continue with the project. The core idea of each quality gate is described below.



A sneak peek into the ML Lifecycle activities we will cover in this course

In the upcoming modules, we'll take a deep dive into each stage of the ML Lifecycle and explore the feedback loops within it. We'll discuss specific activities in each stage that will improve your ML processes. Try to identify the activities that you want to focus on.



Practical recommendations

If you're uncertain about what to do next, consider these practical recommendations.

A common discussion in ML teams is which activities of the ML Lifecycle to focus on in each project phase, especially the Exploration Phase. Have this discussion in the planning phase based on the requirements of the ML system.

When defining the Exploration, Implementation, and Productisation phases, you should decide which project management framework you want to use. We like to use Kanban for Exploration and SCRUM for Implementation and Productization.

Make deliberate choices regarding the technical debt you accept during the Exploration Phase.
Accumulating debt may speed up model development, but remember, it must be addressed.

Self-assessment of comprehension

Take a moment to answer the following questions about the content covered in this module. Keep in mind that there's only one correct answer for each question.

1. Which of the following are NOT part of the MLOps Principles?

- A) Scalable, Trustworthy, Balanced
- B) Reproducible, Accountable, Continuous
- C) Scalable, Continuous, Reproducible

2. What phase is not part of Data Engineering?

- A) Data Cleaning
- B) Data Ingestion
- C) Data Management

3. What are the number of phases in the ML Lifecycle?

- A) 7
- B) 8
- C) 9

4. What are the first letters of the ML Principles?

- A) S, R, A, P, T, C
- B) S, C, A, C, T, C
- C) S, R, A, C, T, C

5. Which accountability should make sure that the following Hidden Technical Debt won't happen: "When planning the project we didn't look into the existing data"

- A) Product owning
- B) Data steward
- C) Data scientist

Self-assessment of comprehension

6. What is the difference between a data engineer and a data scientist?

- A) A data engineer is responsible for the Data Management Phase of the lifecycle while the data scientist is responsible for the Data Preparation and Data Ingestion Phase.
- B) A data engineer prepares the data for the ML models while the data scientist creates the models based on that data.
- C) A data engineer does generic data preparation, while a data scientist does case-specific data preparation.

7. Which of the following processes are the supporting activities in the ML Lifecycle?

- A) Software Design and Development
- B) Software Design and Management
- C) Software Management and Software Monitoring

8. What is the most accurate statement about the ML Principles and the appliedAI Project Phases Framework?

- A) All ML Principles should be put into effect by latest at the end of the Productization Phase.
- B) The ML Principles should be put into effect by latest at the end of the Exploration Phase.
- C) The ML Principles should be put into effect throughout every stage of the Project Phases Framework.

Self-assessment of comprehension

9. What is the most accurate definition of the concept of "Hidden Technical Debt"?

- A) Hidden Technical Debt refers to the maintenance problems that arise with machine learning systems due to the accumulation of costs incurred by moving quickly during development and deployment.
- B) Hidden Technical Debt refers to the technical errors that occur when developers use outdated or unsupported software libraries, which may be paid off by updating software regularly and using only the latest stable versions.
- C) Hidden Technical Debt is the term used to describe code that developers intentionally hide from their colleagues and managers to avoid scrutiny and criticism.

10. What three concepts are missing in this definition of MLOps:

"MLOps comprises a set of tools and _____ to support progress through the ML project _____. The goal is to create systematic ways to think about data, modeling and _____, while supporting the best practices."

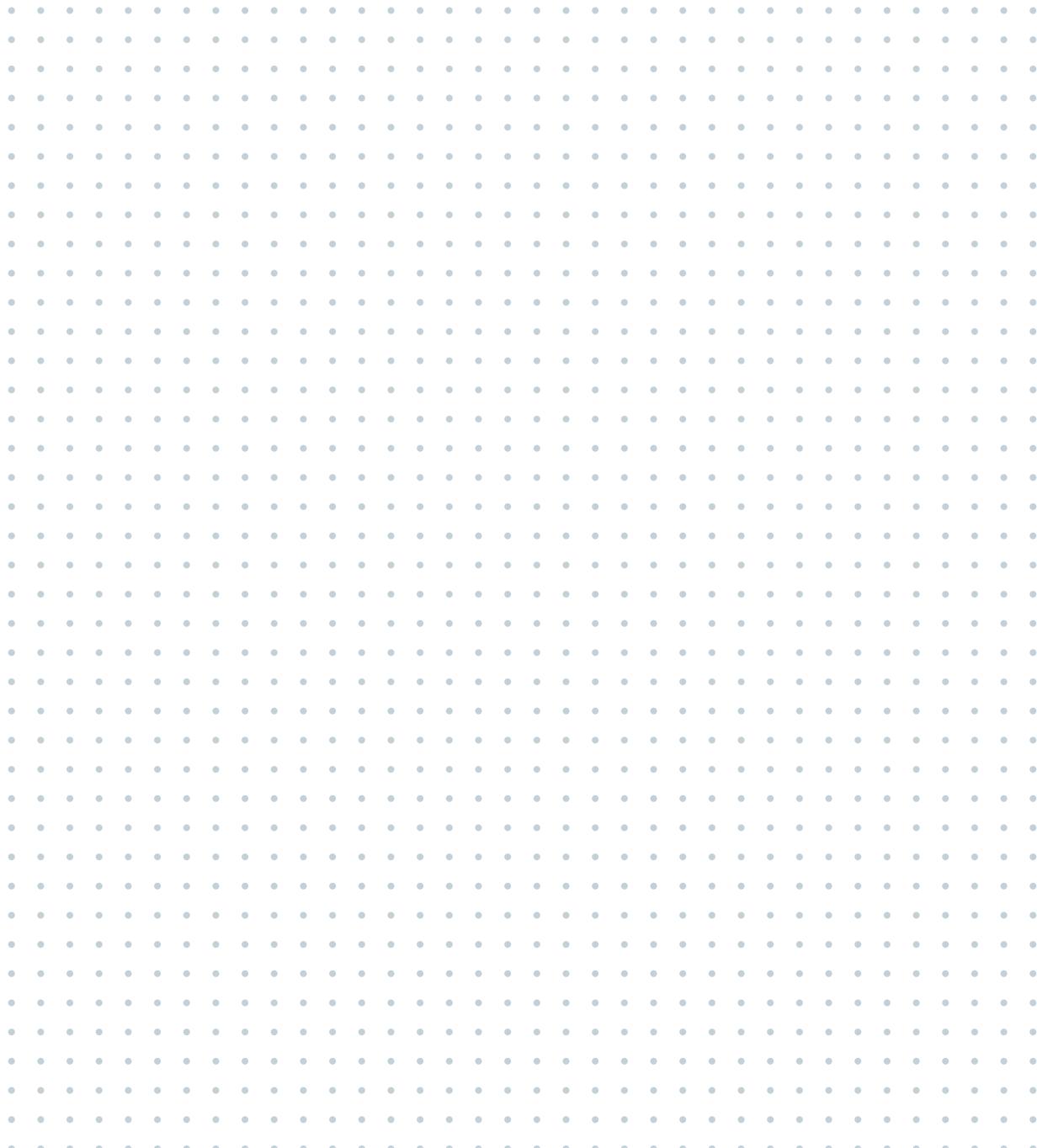
- A) practices, accountabilities, software design
- B) principles, lifecycle, deployment
- C) processes, workflow, maintenance

11. What does a feasibility report look like that should be delivered at the end of the Exploration and Implementation Phase of the appliedAI Project Phases Framework?

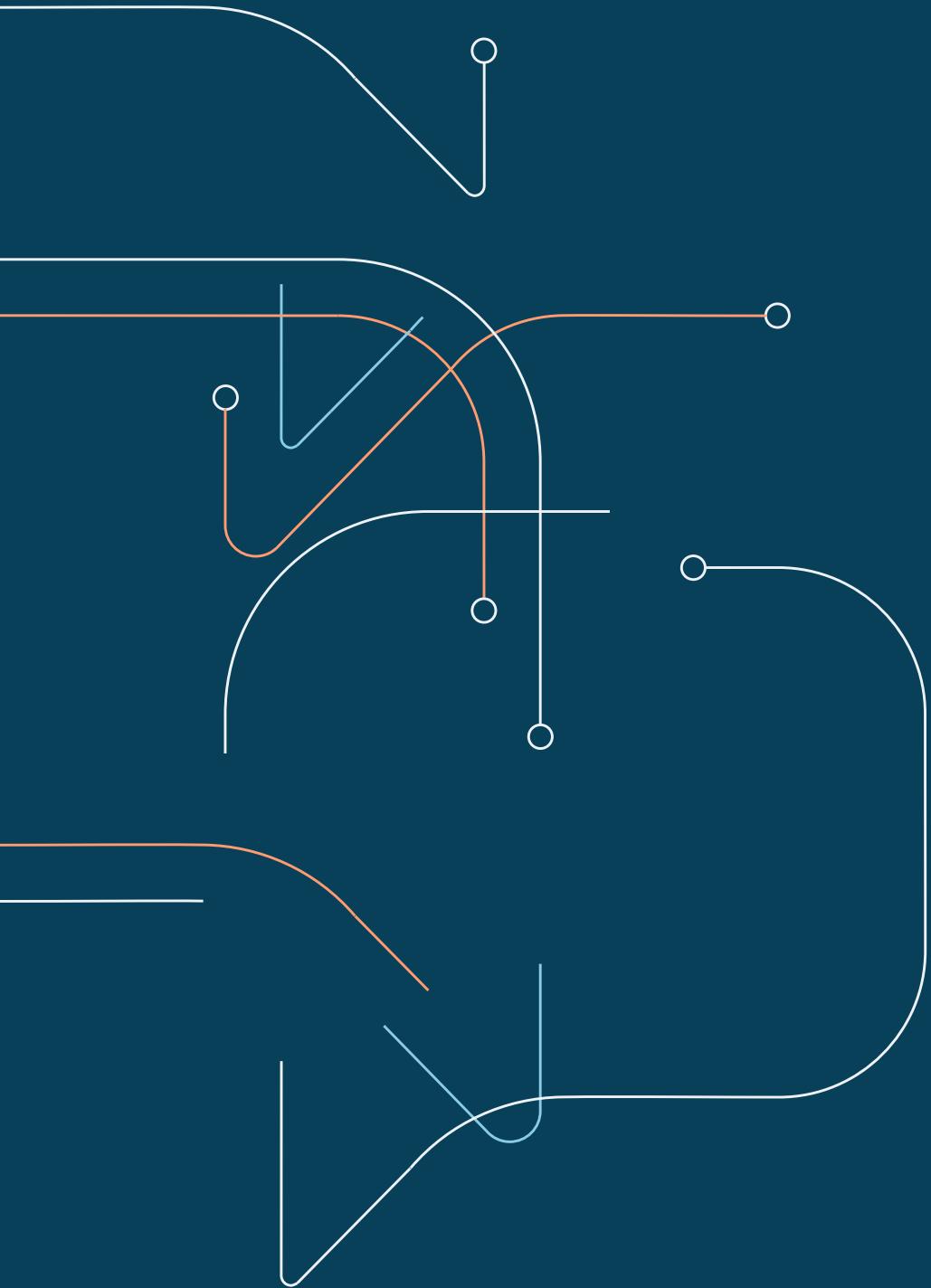
- A) It is a document that reports on how the project phase was conducted.
- B) It is a document assessing the technical feasibility of the success in the next project phase.
- C) It is a document that contains the Model and Data Cards of the model that were trained in both phases of the project.

Module review

Each module begins and ends with this page. At the beginning of the module, we asked you to write down some questions that you would like answered at the end. Once you have written down these questions and worked through the content, take some time to look at the questions again and explain the answers to yourself.

A large grid of grey dots on a white background, intended for handwritten notes. The grid consists of approximately 20 columns and 30 rows of dots, providing a structured area for writing.

02 Scoping



Intended learning objectives

In this module, our focus is on planning an ML project. We will focus on four project workshop topics and highlight the importance of creating a shared document during the Scoping Stage.

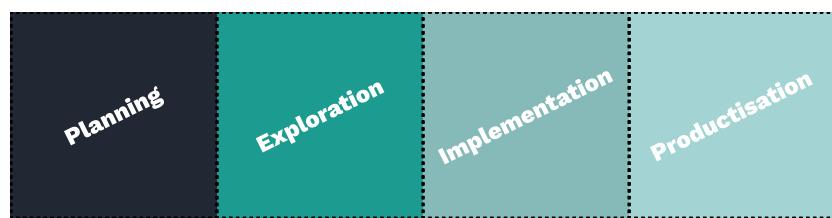
By the end of this module, you will have developed the following proficiencies:

- ✓ Describe the importance of holding a project planning workshop before diving into coding.
- ✓ Explain the advantages of establishing common knowledge within an ML team.
- ✓ Describe the topics to be covered in a project planning workshop.
- ✓ Name the key considerations that need to be discussed for each topic in a project planning workshop.
- ✓ List at least five system requirements of a ML system.
- ✓ Explain under what circumstances it is appropriate to end a project after the Scoping Stage.

Before you begin, we recommend that you take a look at the contents of the module. Take a few minutes to go through each page and look at the headings and visuals. Try to get an overview of the module's content. When you are done, go to the last page of the module and write down a few questions you would like to have answered at the end of the module. We will ask you to review these questions later after you have worked through the content.

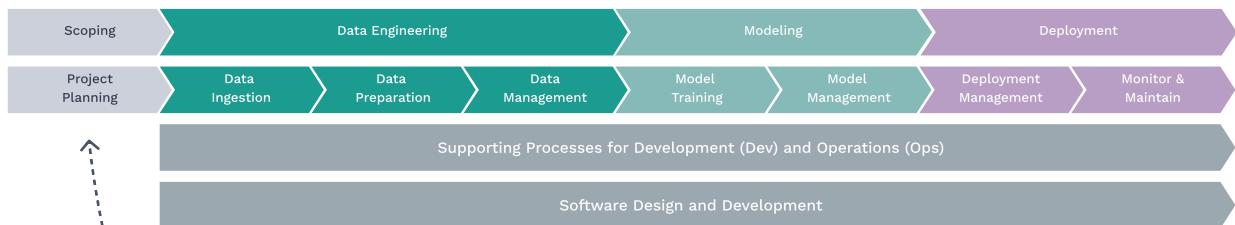
Introduction to the Scoping Stage

The first step in the ML Lifecycle is Scoping, which sets the course for the entire ML project. During this phase, the team carefully plans the project, often through a workshop, to identify the requirements of the ML system and the necessary project setup.



1

In this module, we will talk about the Scoping Stage which overlaps with the Planning Phase of the Project Planning Framework

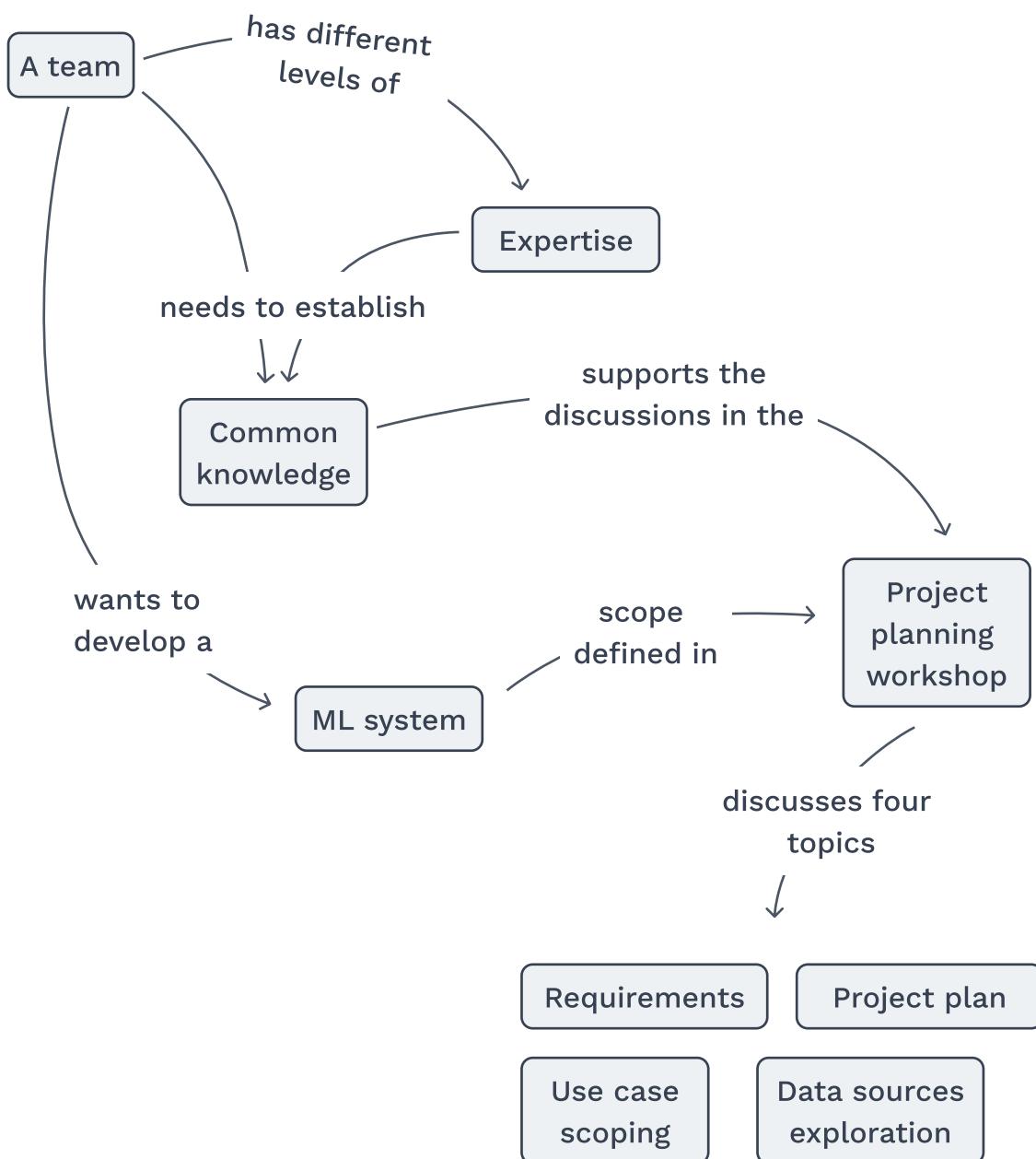


2

During the Planning Phase, the team establishes the requirements of the ML system and sets the trajectory for the project

Do some planning before diving into coding

Jumping straight into coding is rarely a good idea. It might solve the wrong problem and create systems that later need substantial corrections. Listed here are a few overlooked considerations in project planning. Consider the potential impacts that could emerge from not addressing these issues as the project unfolds.

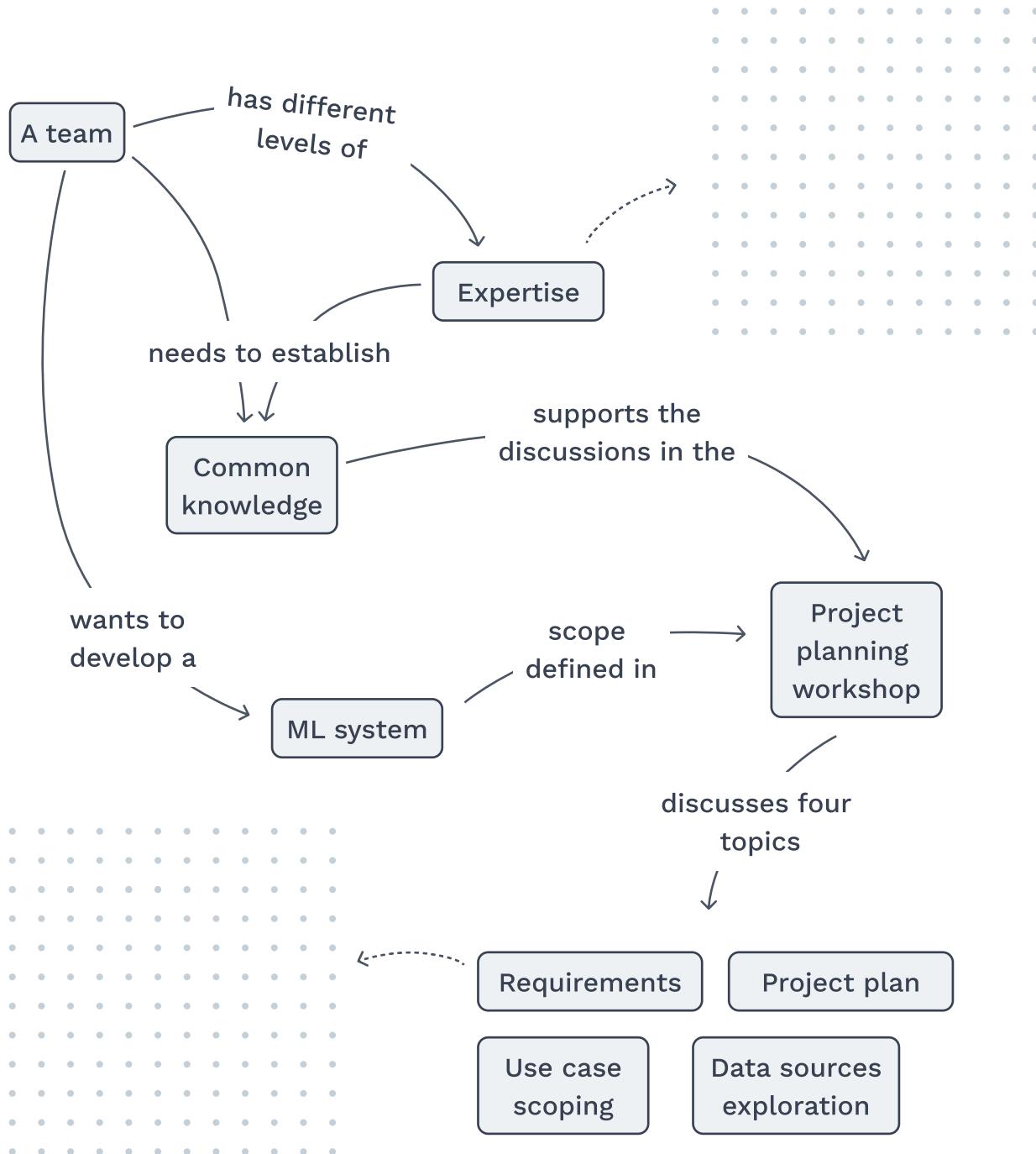


Resources

<https://dl.acm.org/doi/abs/10.1145/3510003.3510209>
<https://arxiv.org/abs/2304.00078>

Stumbling blocks in the planning process

Look again at the planning process, especially the “Expertise” and the “Requirements”. Try to identify stumbling blocks for both topics that could hinder the planning process.



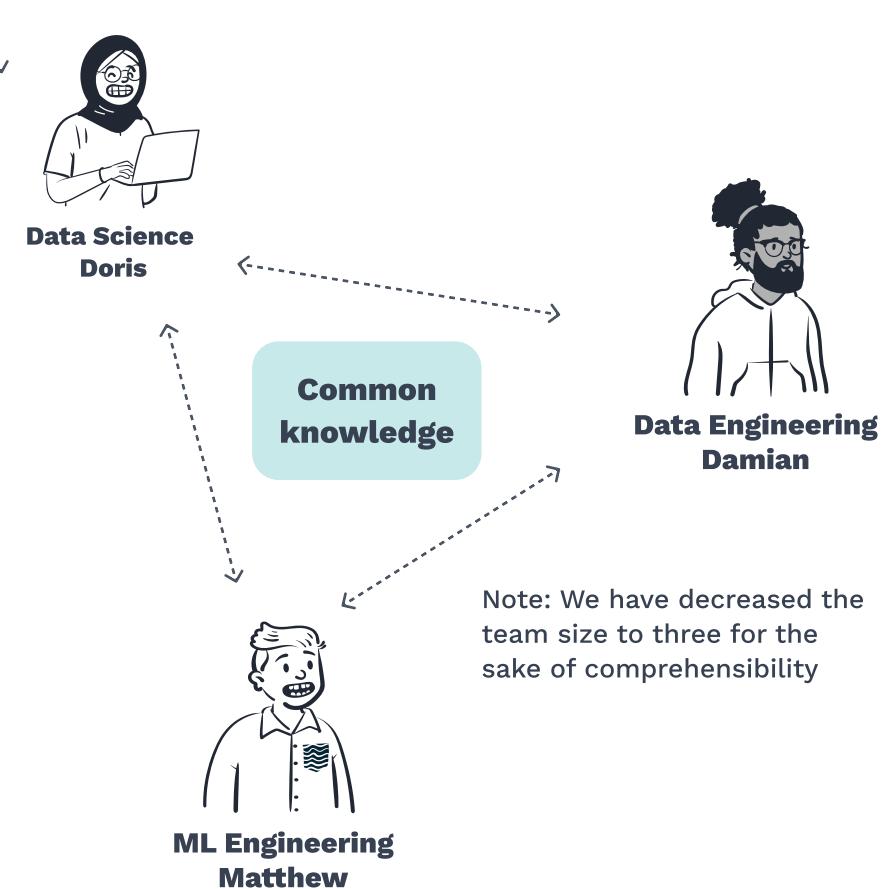
Resources

<https://dl.acm.org/doi/abs/10.1145/3510003.3510209>
<https://arxiv.org/abs/2304.00078>

Creating common knowledge is key for project planning

Fostering common knowledge within a team can minimize the impact of individual blind spots and promote transparency. Common knowledge is established when everyone on the team has reached a minimal understanding of each others work.

“I know that everyone else knows that ML models are probabilistic in nature”



Resources

<https://www.pnas.org/doi/10.1073/pnas.1905518116>

Common knowledge from the perspective of the accountabilities

Here are some examples of contributions to common knowledge that each accountability can make. Take a moment to assess your team and identify any missing contributions that have caused problems in your previous ML projects.



**Solution Architect
Samantha**

“The software infrastructure imposes limitations on the tools that can be used for ML systems.”



**Data Engineering
Damian**

“One of the most time-consuming aspects of machine learning is accessing and preparing the data.”



**Product Owning
Paige**

“When specifying requirements, it is important to consider system requirements such as explainability and latency.”



**ML Engineering
Matthew**

“Even after deployment, machine learning systems typically require significant resources to handle a long tail of ongoing maintenance and updates.”



**Data Science
Doris**

“ML models are probabilistic in nature and may occasionally make inaccurate predictions.”



**Data Steward
Deborah**

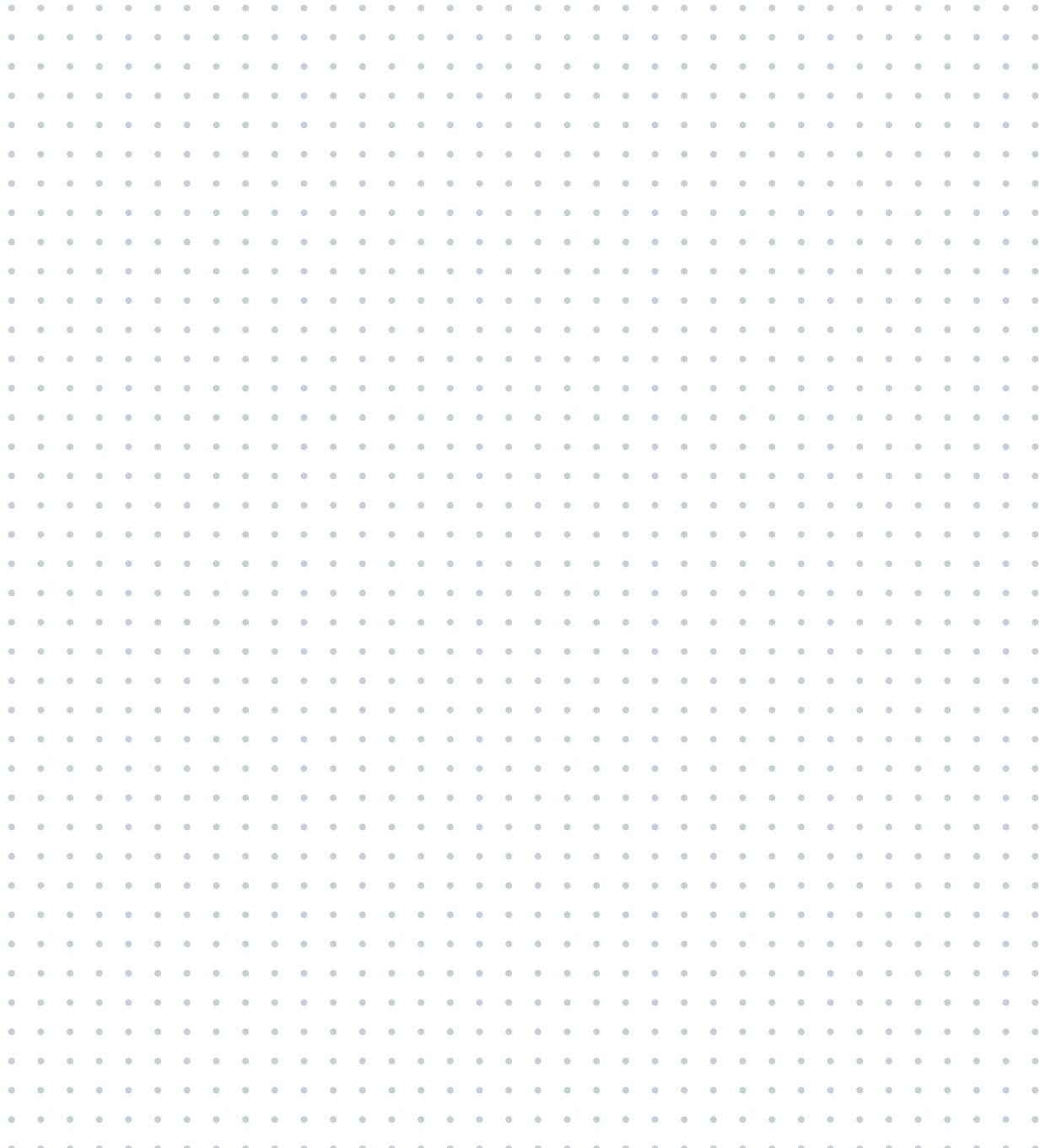
“Data quality guidelines need to be followed within a use case.”

Resources

<https://www.pnas.org/doi/10.1073/pnas.1905518116>

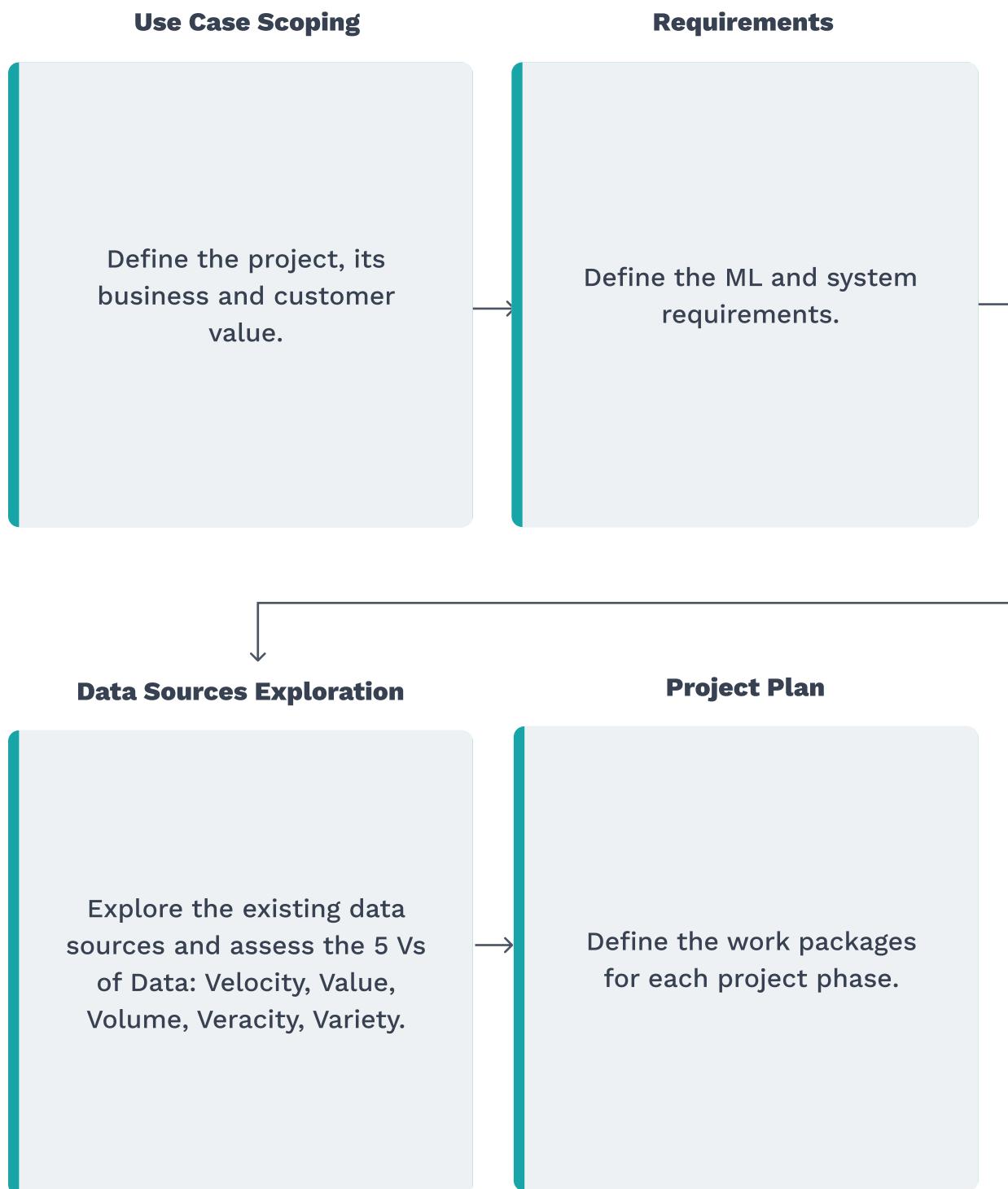
What is your contribution to common knowledge?

Consider your role and accountabilities within your team. What is the common knowledge that you should establish among team members during project planning?



The four topics of the project planning document

The document as a central artifact should cover these four topics. The information on each topic helps the team structure and manage the rest of the project.



Preparation checklist for a successful project planning workshop

To effectively plan your ML project, it's important to have some information in place beforehand. The following checklist outlines key considerations to review before your project planning meeting with your team and all relevant stakeholders and subject matter experts.

Use Case Scoping

We assessed both the business value and implementation feasibility of the chosen use cases.

If required, we have assessed the regulatory requirements.

Requirements

We have defined a list of requirements with all relevant stakeholders.

Data Sources Exploration

The team put together a list of the data sources that could be used for the ML system.



Project Plan

We've invited all necessary stakeholders to the meeting, including subject matter experts.

Our budget includes the costs after the deployment of the ML system.

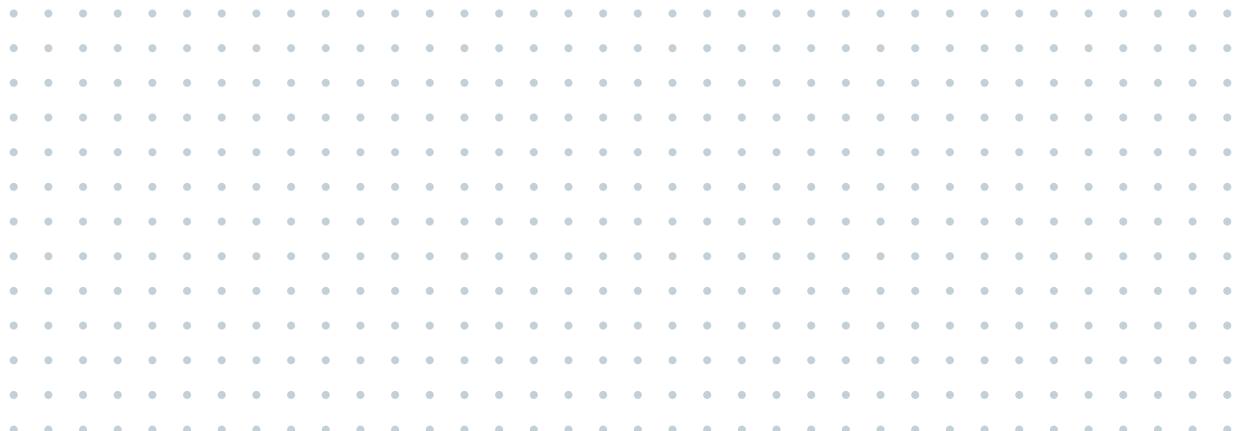
The developer team has a set of coding conventions prepared for discussion.

An exemplary timeline for a project planning workshop • The first half

Below is an example of a planning workshop that covers the four agenda points. It serves as a guide for the key topics that should be addressed during the workshop. The specific times depend on your specific use case at hand.

Time	Agenda Point	Description
09:00 - 09:15	Introduction	<ul style="list-style-type: none">Take a moment for a brief introduction round
09:15 - 09:45	Use Case Scoping	<ul style="list-style-type: none">Provide an overview of the ML projectDefine the objective of the use case to be implementedIntroduce the project planning document
09:45 - 10:00	Break	<ul style="list-style-type: none">Take a break and recharge
10:00 - 12:00	Requirements	<ul style="list-style-type: none">Select the metrics to assess the model's performanceEstablish the baseline to test the model's performanceSpecify the minimum acceptable threshold for these metricsDefine any additional system requirements, such as latency and security

Try to go through the agenda items and identify which of these ideas you would like to implement in your next project planning workshop:



An exemplary timeline for a project planning workshop • The second half

Time	Agenda Point	Description
12:00 - 13:00	Lunch Break	<ul style="list-style-type: none">• Give yourself a break and refuel with some food
13:00 - 14:30	Data Sources Exploration	<ul style="list-style-type: none">• Examine the data available for the ML system• Assess its quality, accessibility, and potential limitations
14:30 - 15:45	Project Plan	<ul style="list-style-type: none">• Define how team members will be trained• Define the project management approach and the project phases• Define the higher level epics per project phase• Assign the ML Accountabilities to the team members and color code them in the list of epics• Define how new team members will be onboarded
15:45 - 16:00	Closing	<ul style="list-style-type: none">• Sum up the main points from the workshop and restate the following actions for the project

Try to go through the agenda items and identify which of these ideas you would like to implement in your next project planning workshop:

Use case scoping

The goal of this workshop session is to define the project and its business implications. The product owner is responsible for defining the scope and must answer four key questions. If you have a use case at hand, feel free to write down your answers to these questions.

What is the project about?

What is the business value for this use case?



**Product Owning
Paige**

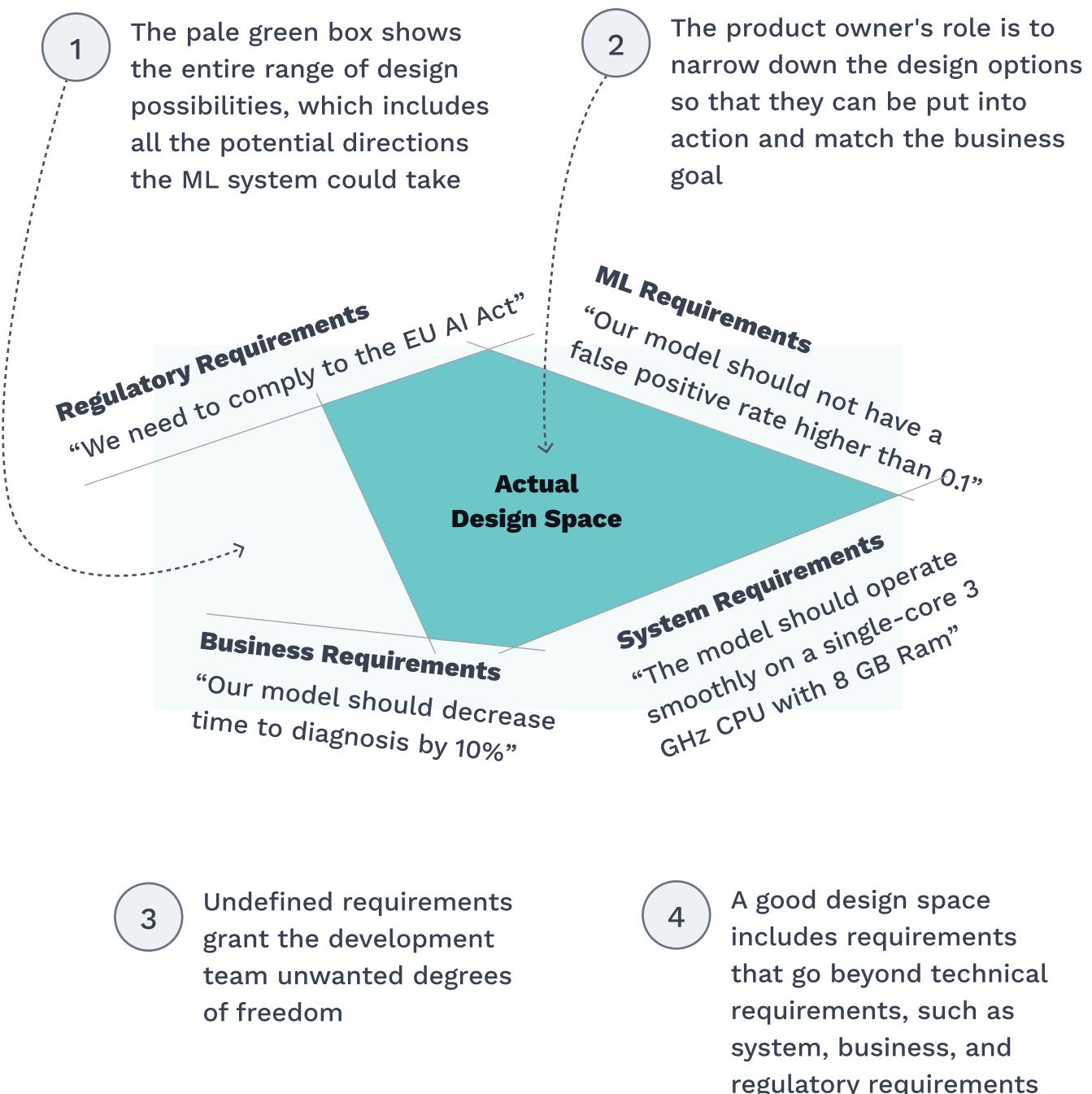
What is the customer value for this use case?

Is AI necessary for this use case?



Requirements engineering is an effort to reduce the possible design space

Given the numerous requirements for each ML system, there are countless paths to explore. Therefore, requirements engineering aims to narrow down the possible design space to the actual design space. Establishing well-defined requirements supports the team in determining the desired type of ML system to be developed.



Resources

<https://www.neverletdown.net/2010/09/exploring-design-space.html>

A list of overlooked requirements

Take a look at the list of requirements that are often overlooked during the planning of ML projects. Use them as inspiration for an upcoming project. Remember that your team will only focus on implementing a few of these requirements. We have arranged the requirements in alphabetical order.

Auto-scaling	Auto-scaling refers to the dynamic adjustment of computational resources to match the changing demands of a model, guaranteeing optimal performance and resource use.
Completeness	An indication of the comprehensiveness of available data, as a proportion of the entire dataset, to address specific information requirements.
Explainability	The extent to which the mechanics of an ML-supported system can be explained in human terms, both the internal processes and the outcomes of the ML system.
Fairness	The ability of a system to operate in a fair and unbiased manner.
Integrity	The ability to make sure that data is real, accurate, and safeguarded from unauthorized modification.
Privacy	An algorithm is private if an observer examining the output is not able to determine whether a specific individual's information was used in the computation.
Reliability	The probability of the software performing without failure for a specific number of uses or amount of time.



continues on
next page

Resources

<https://arxiv.org/pdf/2203.11063.pdf>

<https://link.springer.com/article/10.1007/s00766-022-00395-3>

A list of overlooked requirements

Reproducibility	One can repeatedly run your algorithm on certain datasets and obtain the same (or similar) results. Also, each phase of data processing, model training, and model deployment should produce identical results for the same input.
Safety	The absence of failures or conditions that render a system dangerous.
Security	Security measures ensure a system's safety against espionage or sabotage.
Traceability	The ability to trace every asset in an ML project across the development lifecycle, for example through lineage.
Transparency	The extent to which a human user can infer why the system made a particular decision or produced a particular externally-visible behavior.
Trust	A trusted system is one relied upon to enforce a specific security policy. It involves the capability to demonstrate the correctness or reasonableness of the output from an ML-enabled system.

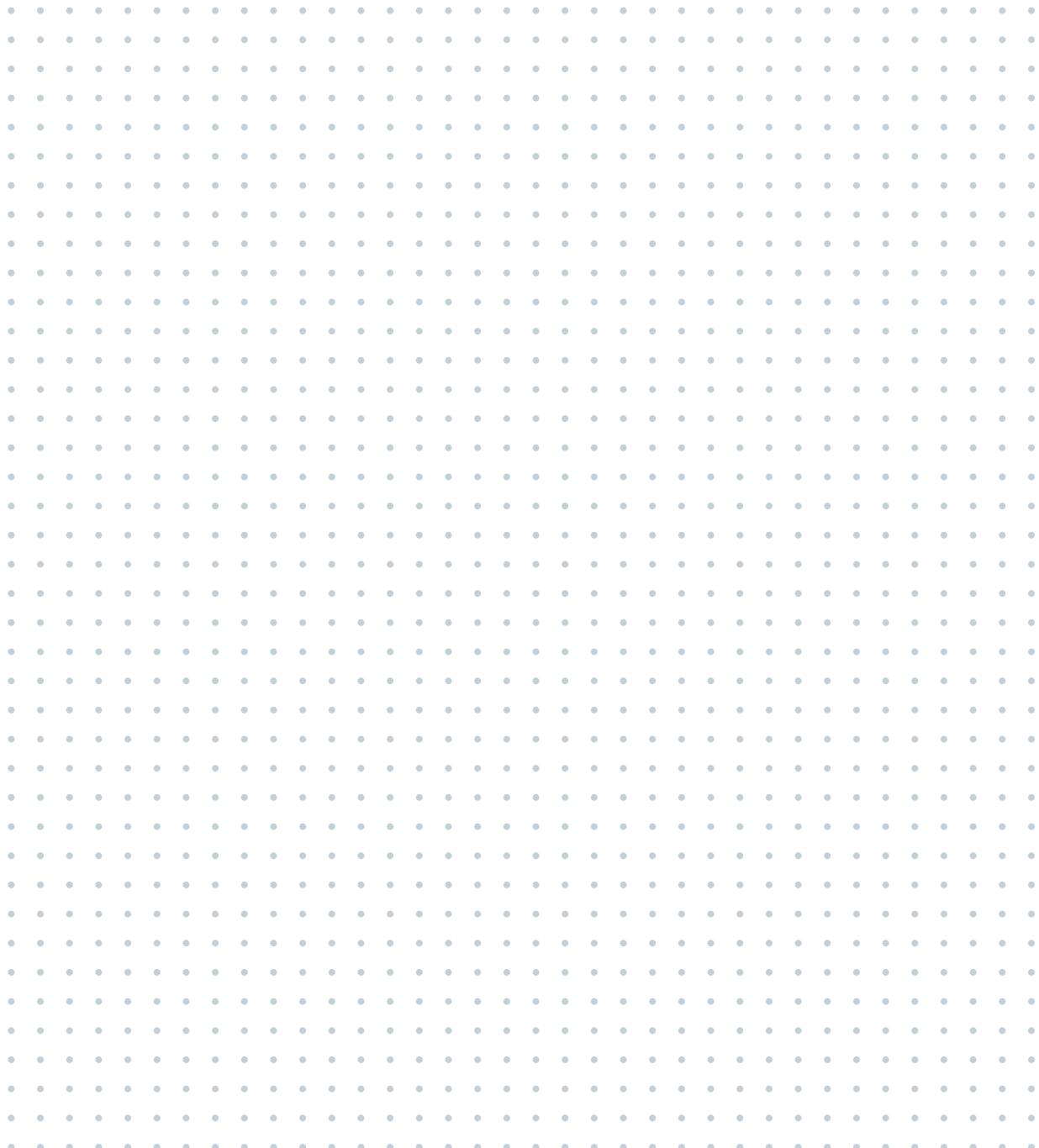
Resources

<https://arxiv.org/pdf/2203.11063.pdf>

<https://link.springer.com/article/10.1007/s00766-022-00395-3>

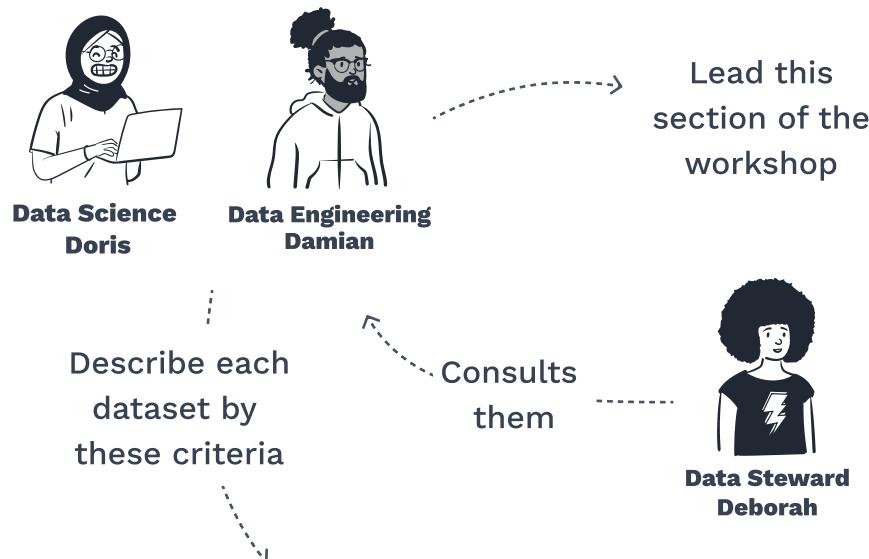
Optimizing your requirements

Pause for a moment and reevaluate one of your recent ML projects. Compare the requirements you set for that project with the ones discussed on the previous pages. Try to identify any requirements that would have been advantageous to include.



Data sources exploration

The goal of this workshop session is to get clarification about the datasets relevant for the use case and its quality and scope.



Description of the dataset:

Data Format:

e.g. csv, txt, parquet

How is this dataset relevant to the current use case?

High

Medium

Low

How easy is it to acquire the data?

Easy

Medium

Hard

How much data do we have?

< 1 million rows

< 10 million rows

> 10 million rows

What is the velocity of the data?

Static

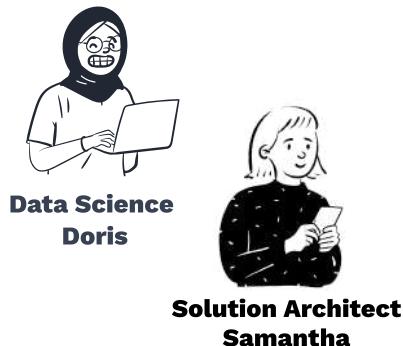
Batch

Stream

Which columns are useful for creating labels or features?

Model implementation details

Once you have identified and described your data sources, you can determine the key features of the models your team intends to build.



Lead this
section of the
workshop

What are the inputs and outputs of the model?

What is the learning type?

Supervised

Unsupervised

Reinforcement learning

Is the solution known?

Yes, easy to replicate

Yes, hard to replicate

No

What is the size of the model?

Small (Baseline)

Pre-trained

Large

Rule-based

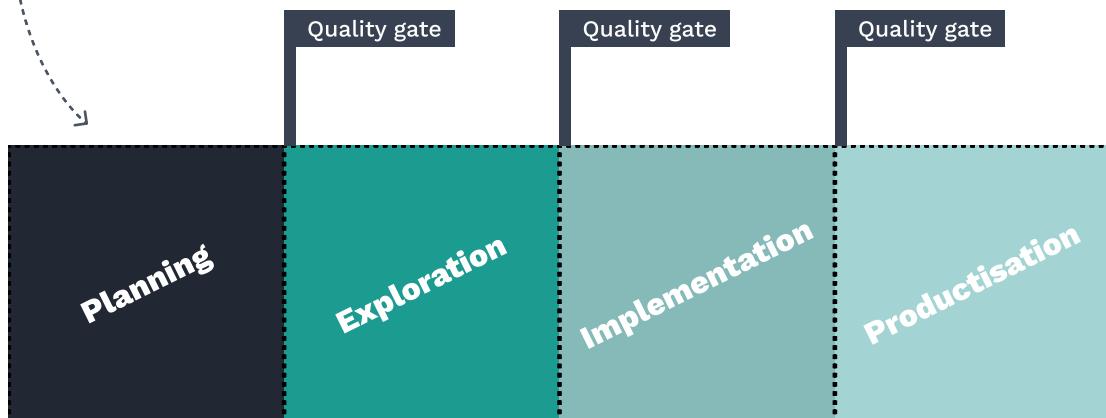
What are the evaluation metrics?

Summary

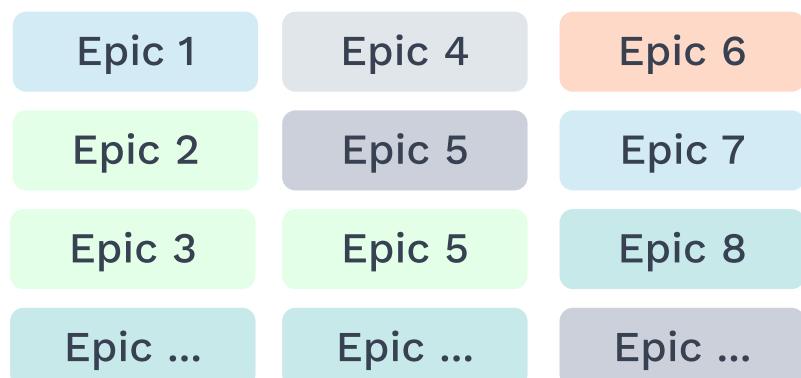
Project planning is epic planning

The goal of the project plan session is to define the high level activities (epics) that need to take place per project phase.

- 1 During the Planning Phase, the team defines high-level epics for the remaining project phases



- 2 Epics are color-coded based on their accountability



Project planning • The quality gates for the Planning and Exploration Phase

This page and the following describe a few key quality checks for the Planning, Exploration, and Implementation Phase. Use them as a guide for your next project.

Planning

We have a committed team with a shared understanding of the system's purpose.

We have clearly defined metrics in place to measure progress and success.

We have access to enough data for testing optimization and data-dependent goals.

We have a comprehensive list of requirements/user stories for our epics.

We have set stop-criteria that determine when the project can move beyond the Exploration Phase.

Exploration

We have identified several promising ways to achieve our optimization and data-dependent goals. These paths adhere to all technical requirements.

We have an infrastructure and technology stack that supports continuous iterations.

We have a clear plan for acquiring and handling the necessary data, if needed.

We have a well-defined list of features to implement, along with estimates or validations of their optimization potential.

We have a functional development setup that fosters smooth communication and collaboration between developers and technical stakeholders.

Project planning • The quality gates for the Implementation Phase

Implementation

We have conducted extensive research and a deep dive into the candidate solutions, enabling us to evaluate if their optimized performance will meet the given requirements.



We have developed a refined backlog for the Productisation Phase encompassing estimates or validations of optimization goals for each item.



We have implemented a running development setup that promotes seamless communication and collaboration between developers and technical stakeholders.



We have established an infrastructure and technology stack that adheres to best practices throughout the ML Lifecycle, facilitating continuous integration and development of ML Models.

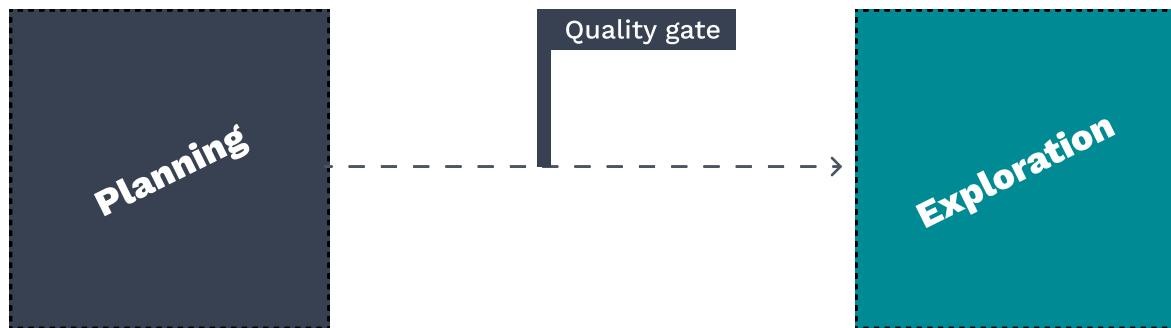


We have assessed the probability of success during the Productisation Phase and determined the level of risk involved. Moreover, we have provided recommendations to the production architecture that best satisfies the requirements of the project.



The planning quality gate

A quality gate serves as a midway assessment to determine whether it is justifiable to continue with the project. Consider the transition from the Planning Phase to the Exploration Phase and come up with a few reasons it might not be reasonable to proceed from Planning to Exploration.



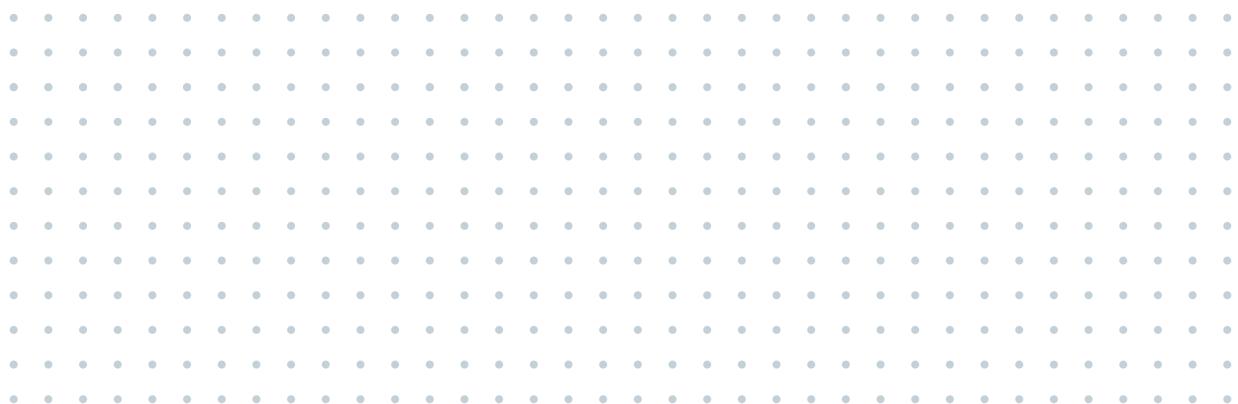
In this phase defines the project for the selected use case and the epics to be implemented in each project phase

In this phase, the team explores the feasibility of the project based on a small dataset with a baseline model

There is not enough data and it is unlikely more data can be generated.

The use case is not valuable in terms of its business value.

Based on these two reasons, try to come up with your own



Practical recommendations

If you're uncertain about what to do next, consider these practical recommendations.

Create and share a document containing the project workshop results with all stakeholders to ensure transparency, alignment, and informed decision-making. A typical format for the document is a (digital) whiteboard.

Define clear quality gates to facilitate the decision-making process for progressing to the next project phase. These quality gates must be highly specific, aligning with the nature of each respective gate, such as focusing on data preparation during the Exploration Phase.

Prepare a list of requirements that go beyond functional aspects such as accuracy. Take careful consideration of the requirements that should be in place by the end of the Productisation Phase.

To improve collaboration within the team, it is crucial to establish a minimum level of common understanding regarding developing ML systems at the project's outset. Make sure that perspectives from each accountability are known to everyone on the team.

Self-assessment of comprehension

Take a moment to answer the following questions about the content covered in this module. Keep in mind that there's only one correct answer for each question.

1. What is the primary purpose of the Planning Phase?

- A) A notebook that discusses the data's potential for the ML model.
- B) Specific use cases that are implemented for the ML project.
- C) A shared document describing the project's scope, its requirements, data sources, and epics.

2. What topics should be discussed in the project planning workshop?

- A) The business value and feasibility of the use cases; the requirements for the ML system; the data sources for the use cases.
- B) The requirements for the ML system; the implementation plan; the data sources.
- C) The data sources; the implementation plan; the algorithms for the ML model.

3. What is common knowledge?

- A) Knowledge that only one team member has.
- B) Knowledge that every individual in the team has.
- C) Knowledge that every individual in the team has and knowledge about the knowledge of all team members.

4. What is the meaning of the word scoping?

- A) The process of defining the boundaries or extent of a project.
- B) The act of examining a project for potential risks and hazards.
- C) The process of tracking a project's progress and status.

Self-assessment of comprehension

5. Under which circumstances should the concept of the false negative rate be common knowledge within a team: “The false negative rate is the proportion of positive cases that are incorrectly identified as negative among all positive cases”

- A) When developing a sentiment analysis model for social media monitoring, missing negative comments can lead to negative consequences for a brand's reputation.
- B) When training a spam detection model, failing to identify all spam emails can lead to the user missing important messages.
- C) When developing a diagnostic model where detecting all positive cases is critical to avoid misdiagnosis and harm to patients.

6. How long is the Planning Phase usually?

- A) 1 to 3 weeks
- B) A few days
- C) 2 to 5 weeks

Self-assessment of comprehension

7. Which of the following examples is a sufficient reason for ending the project after the Planning Phase?

- A) It is impossible to judge if the use case will make it into production.
- B) There is a lack of clarity about the specific requirements for the ML system to be developed.
- C) The model's current accuracy is 60%; we are not sure if we can optimize to 90%.

8. Which of the following activities should be done before the planning workshop takes place?

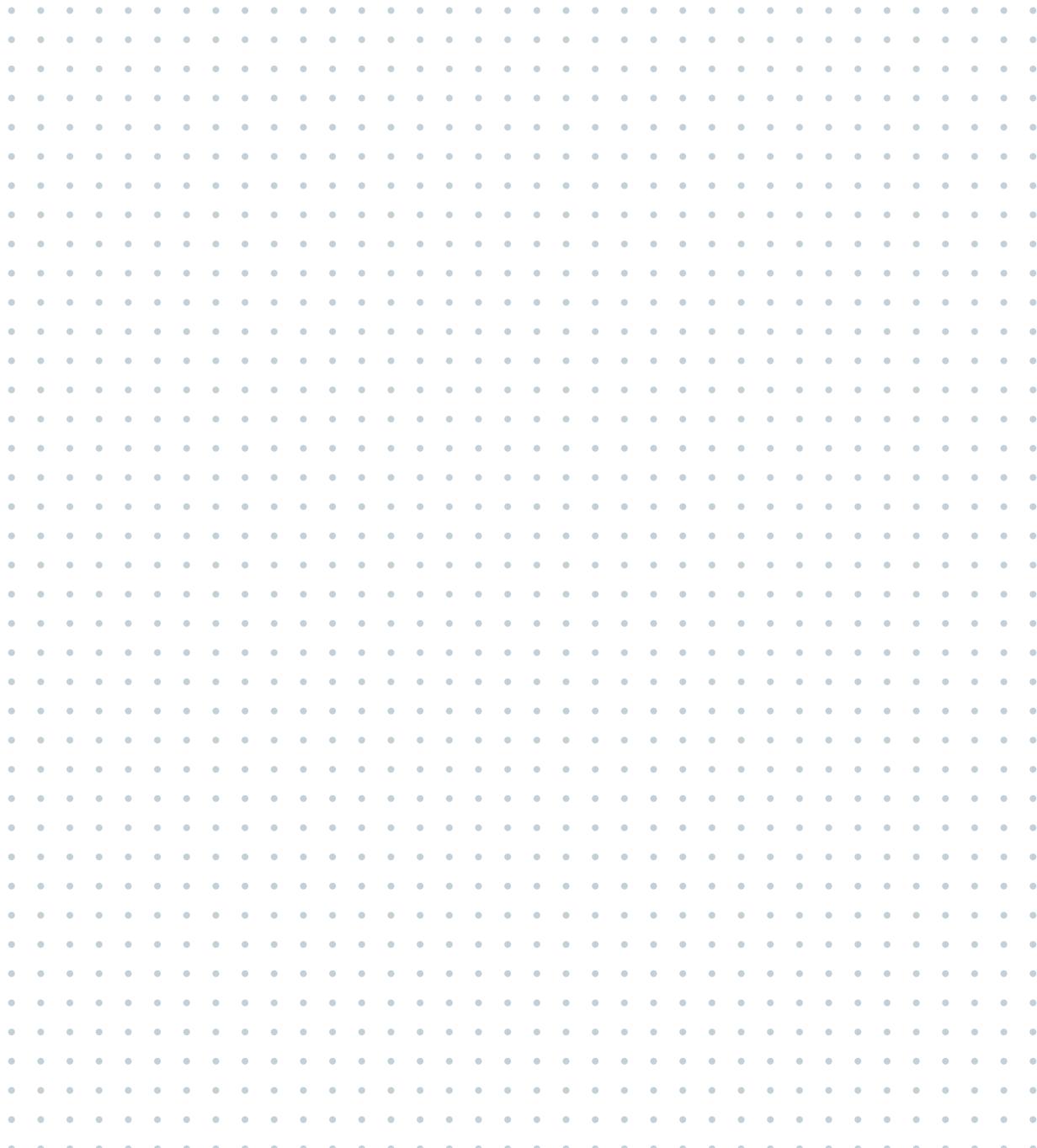
- A) The metric(s) against which the model will be evaluated should be defined.
- B) An overview of the relevant data sources should be created.
- C) The duration of the project phases should be defined.

9. Try to assign the following contribution to common knowledge to the appropriate accountability: “Model performance and accuracy can deteriorate over time, necessitating continuous monitoring and periodic retraining of machine learning systems.”

- A) ML engineer
- B) Software engineer
- C) Data engineering

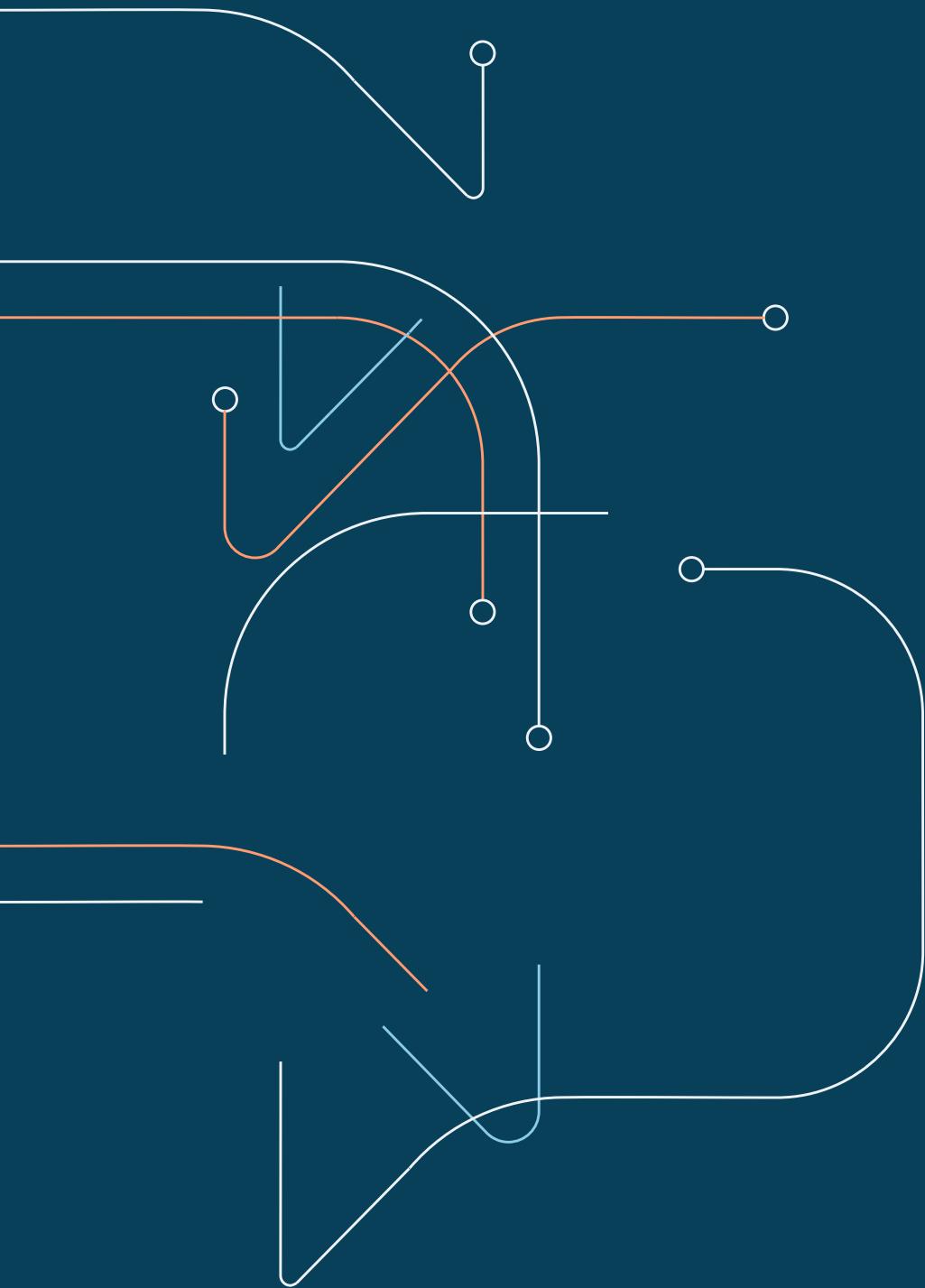
Module review

Each module begins and ends with this page. At the beginning of the module, we asked you to write down some questions that you would like answered at the end. Once you have written down these questions and worked through the content, take some time to look at the questions again and explain the answers to yourself.

A large grid of grey dots on a white background, intended for handwritten notes. The grid consists of approximately 20 columns and 30 rows of dots.

03

Data Engineering



Intended learning objectives

In this module, our focus is on improving data management. We'll start by understanding different data architectures for storing data, then discuss choosing between ELT and ETL patterns. We'll also see how to ensure data quality while ingesting it and how versioning helps with reproducibility. Lastly, we'll explore data catalogs and data lineage, which bring transparency to data within the organization.

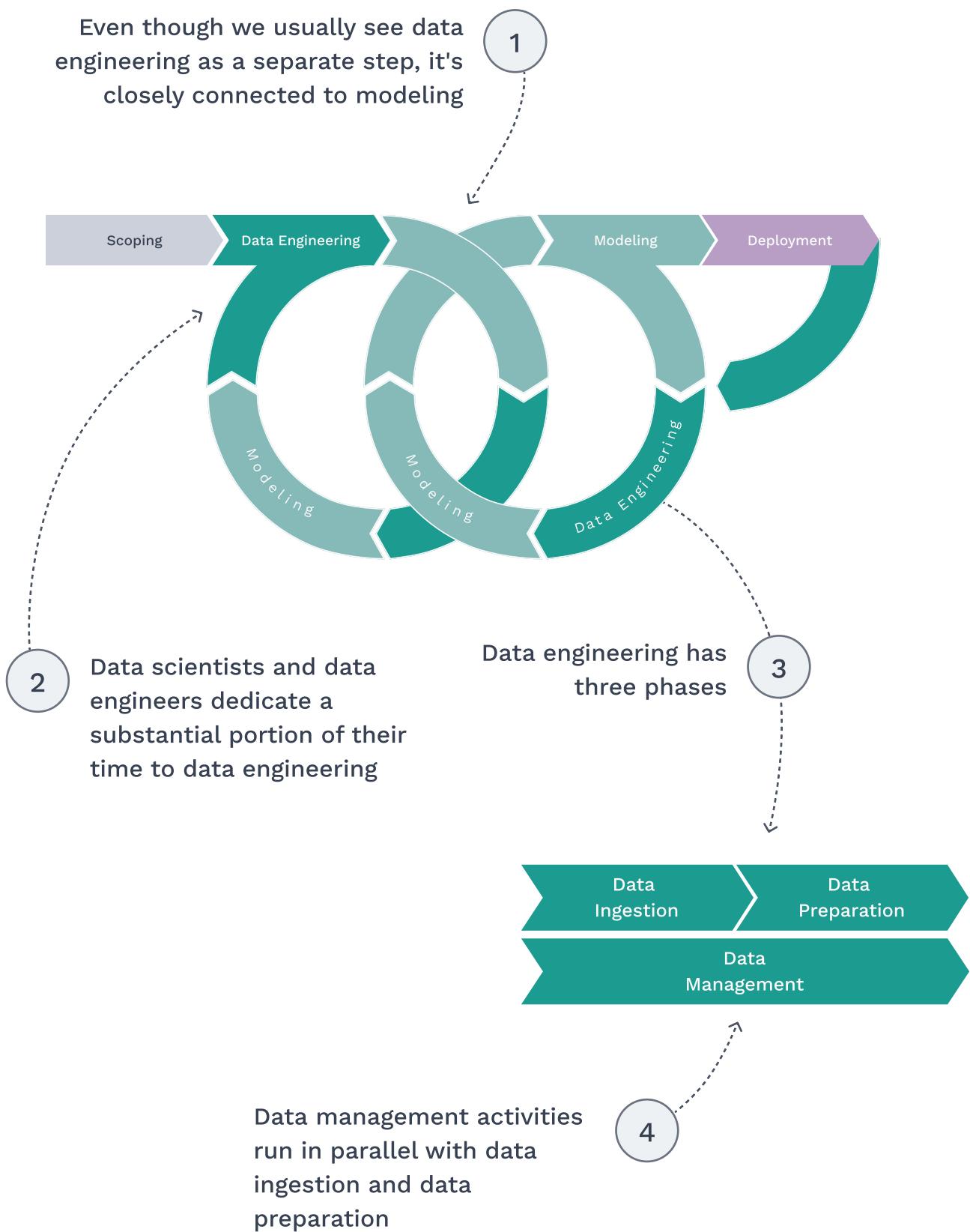
By the end of this module, you will have developed the following proficiencies:

- ✓ Describe the core idea of the data architectures data warehouse, data lake, and data lakehouse.
- ✓ Differentiate Extract, Transform, Load (ETL) from Extract, Load, Transform (ELT).
- ✓ Describe the three types of data versioning architectures.
- ✓ Explain the need for data quality checks in data engineering.
- ✓ Name the benefits of a data catalog from the perspective of the ML Accountabilities.
- ✓ Describe how data catalogs and data lineage improve communication about data within an organization.

Before you begin, we recommend that you take a look at the contents of the module. Take a few minutes to go through each page and look at the headings and visuals. Try to get an overview of the module's content. When you are done, go to the last page of the module and write down a few questions you would like to have answered at the end of the module. We will ask you to review these questions later after you have worked through the content.

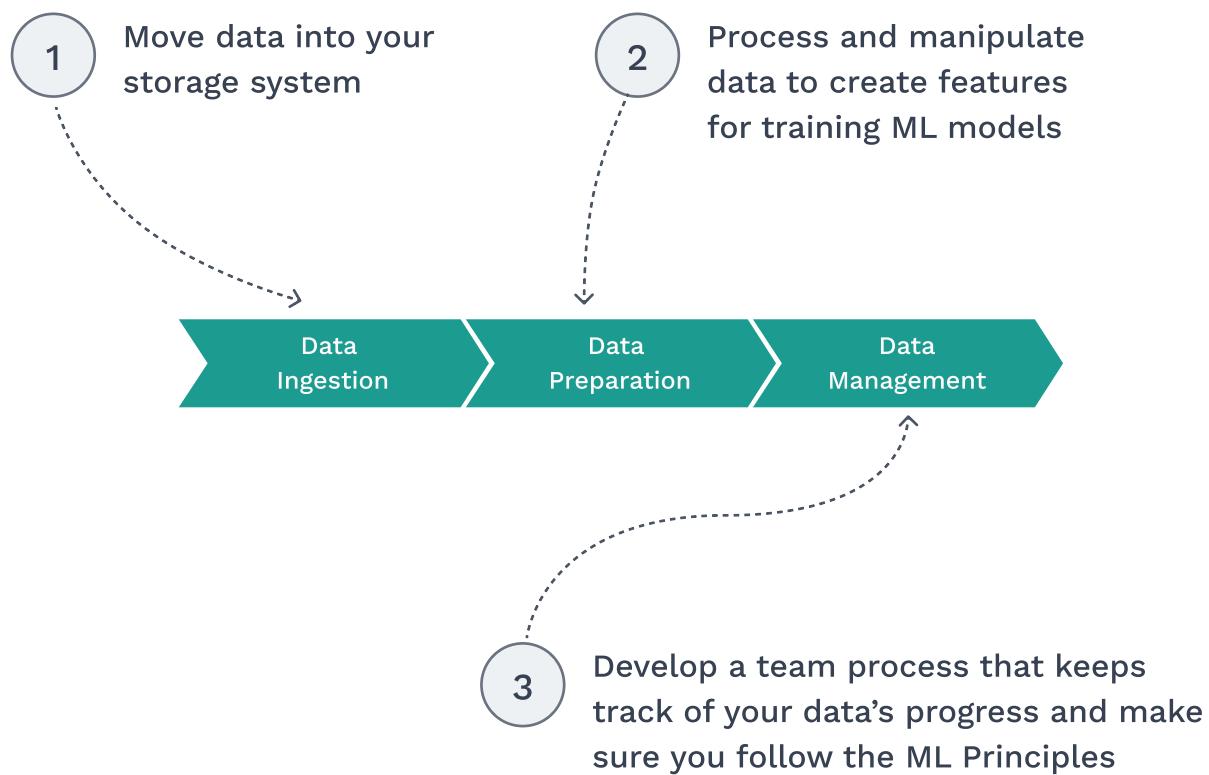
Introduction to the Data Engineering Stage

The main goals of data engineering are: preparing your data for modeling and managing it in a way that follows the ML Principles



The goals of the three Data Engineering Phases

Below you can find a breakdown of the key goals of each Data Engineering Phase.



Typical challenges in data engineering

Take a look at the common challenges that often arise during the Data Engineering Stage. Assess whether you came across any of these challenges in your own work.

The data is not available on a shared infrastructure.

Different people have different access rights to the data.

Intermediate datasets can't be reproduced.

Debugging errors in the data processing is time consuming.

A lot of datasets are not searchable or not findable.

It is challenging to find out if a dataset can be used.

Data documentation is not comprehensive enough to understand the data.

The data team does not get enough time with people who know the data to gain a thorough understanding of it.

The quality of datasets is not automatically checked.

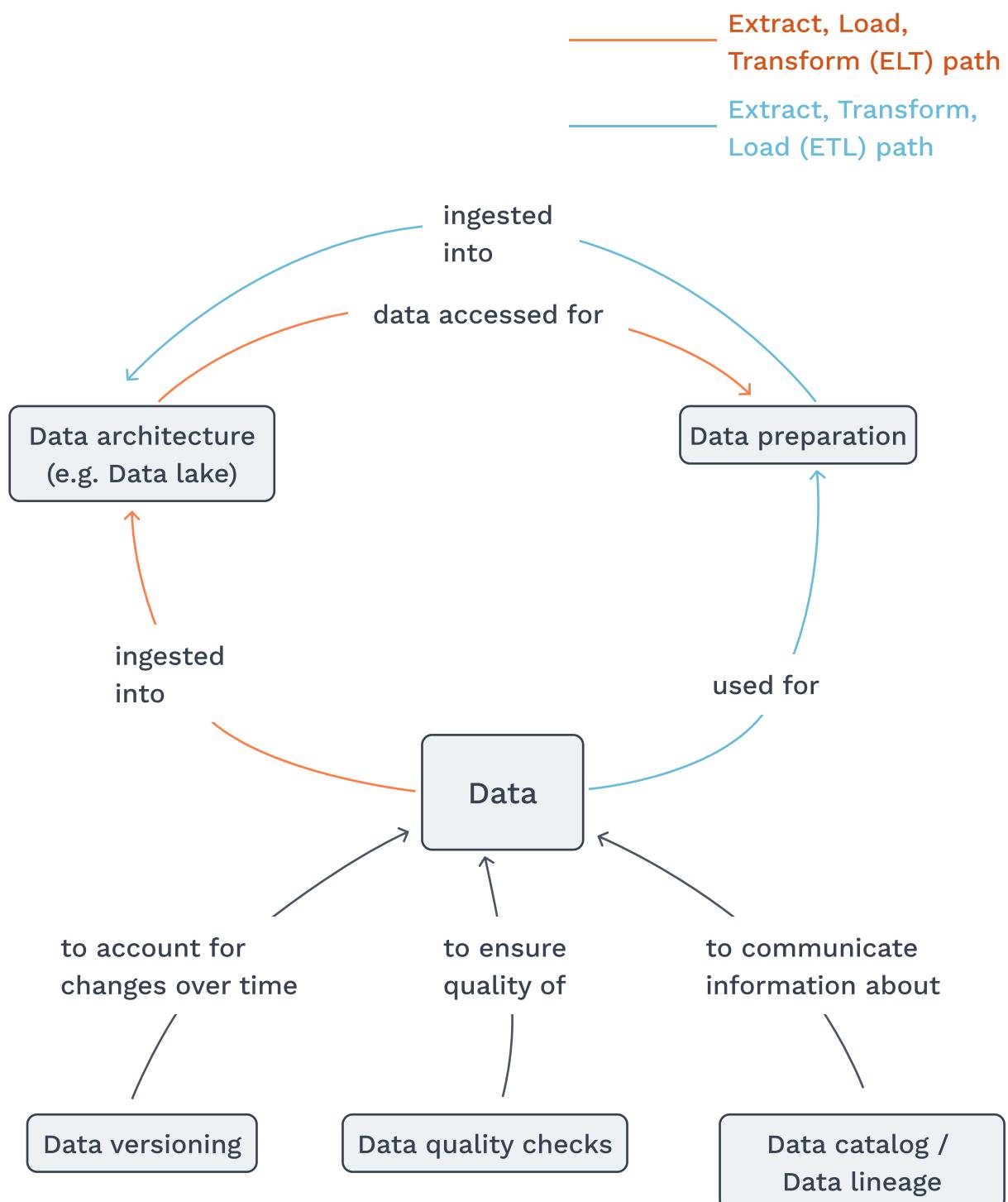
It is not possible to easily reproduce previously used datasets.

Resources

<https://link.springer.com/article/10.1007/s10664-021-09993-1>
<https://arxiv.org/pdf/2007.14130.pdf>

The lifecycle of data

To avoid most of these challenges, data should go through a series of lifecycle activities that each solve a different type of challenge. The following is the data engineering workflow we are going to discuss in the pages ahead.



Data architectures: data lake, data warehouse, and data lakehouse

When managing data, your company or department needs to make informed choices about the data architecture that will be used for data storage.

Generally, three architectures can be discerned. Read the brief descriptions of each and make note of any new features and differences between the architectures that you want to remember.

Data lake

A data lake is a scalable storage system that houses raw data in a variety of formats: structured, semi-structured, and unstructured data, supporting exploratory analytics.

Data warehouse

A data warehouse is a centralized repository that stores structured data, allowing for efficient querying and analysis.

Data lakehouse

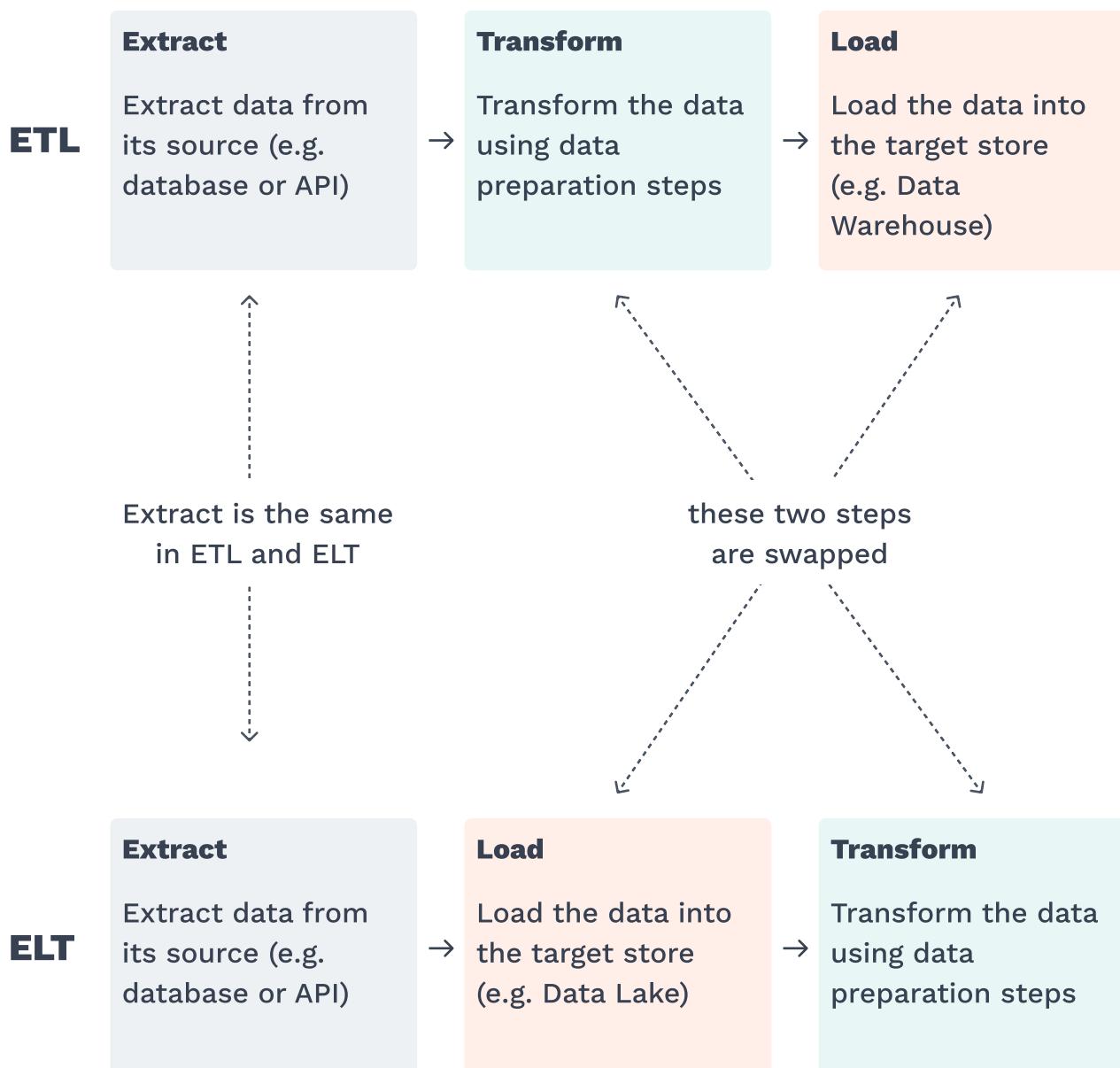
A data lakehouse integrates structured and unstructured data in a unified platform, enabling comprehensive analytics and decision-making with querying capabilities.

Resources

<https://www.databricks.com/blog/2020/01/30/what-is-a-data-lakehouse.html>

ELT and ETL

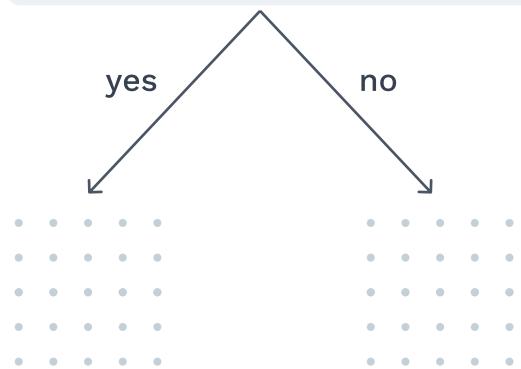
When selecting a data architecture, your team should also consider whether an ELT or ETL approach to data preparation is needed. Both concepts refer to when the data in your workflow undergoes transformations. There are no definite right or wrong answers, but certain approaches may work better for specific use cases.



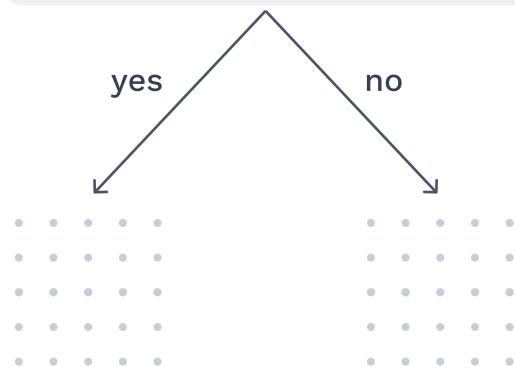
ELT/ETL decision boundaries

Take a look at the decision trees provided below. Each tree has a node and two edges. Your task is to figure out which pattern (ETL or ELT) fits best with each tree.

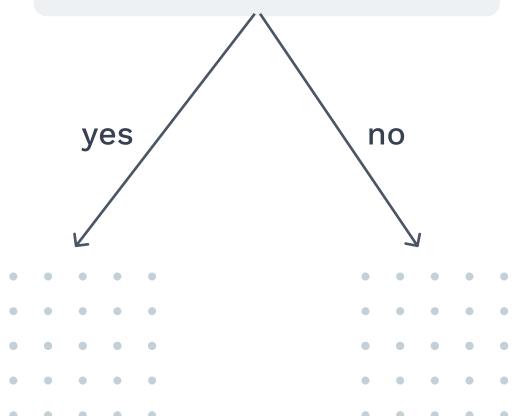
There is a need to reproduce
the results of your data
preparation steps



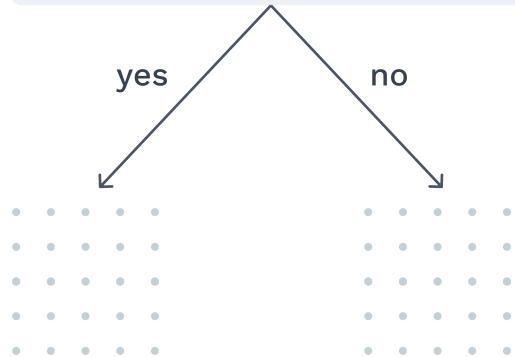
There is a need to reproduce
your models



You have limited storage capacity for your data

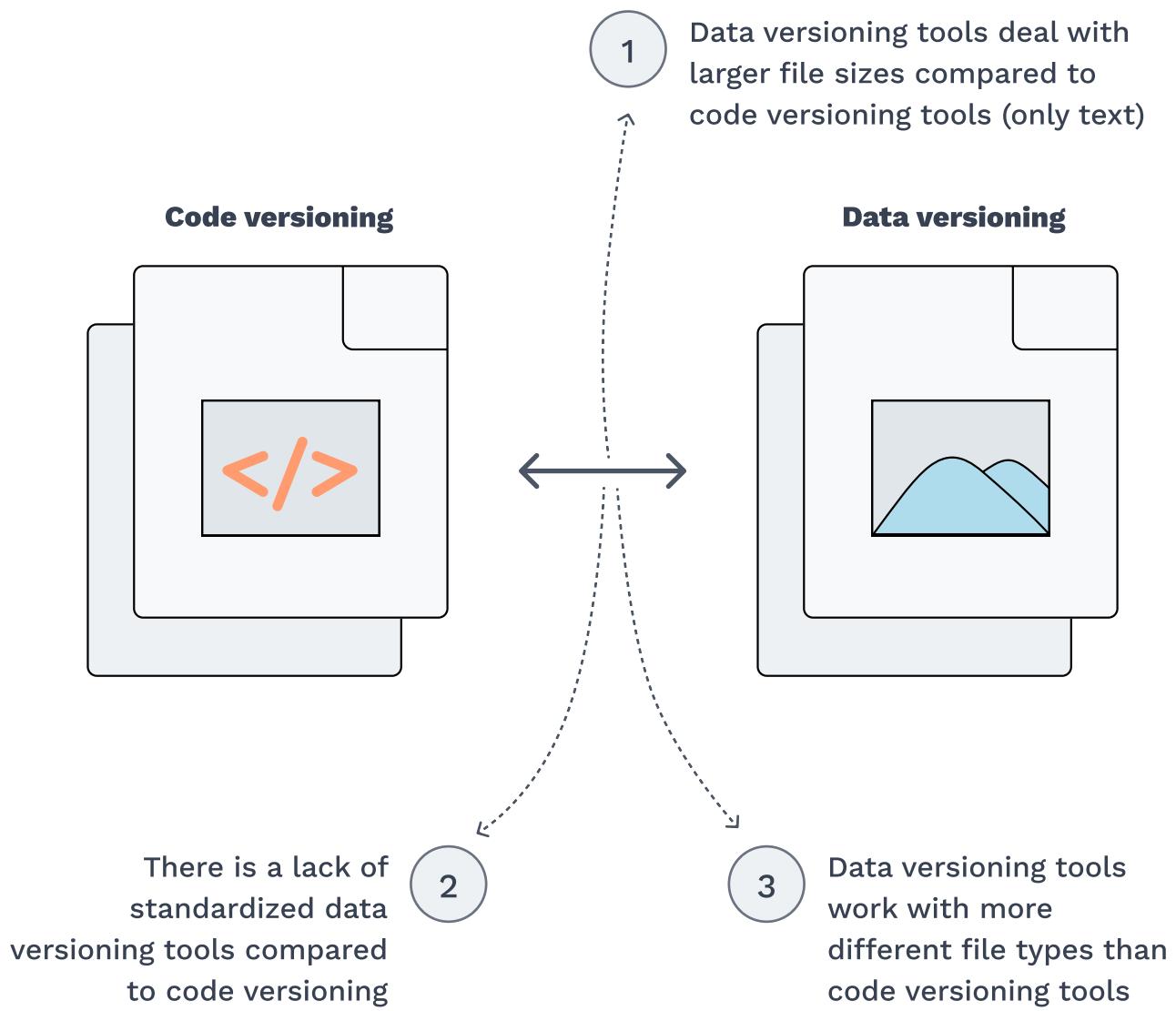


We have control of the data source



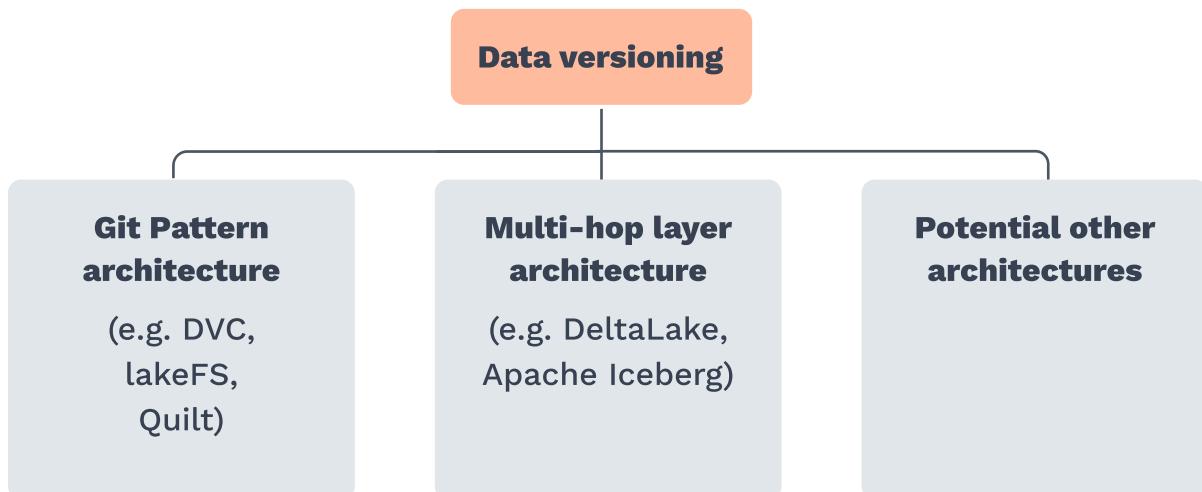
Why versioning data is different from versioning code

Once you have your data architecture in place and have decided between ELT or ETL, it's crucial to track the versions of the data. Many developers try to understand data versioning by comparing it to code versioning like Git. Although they share similarities, data versioning is different than code versioning. Below are three aspects in which data versioning and code versioning differ.



Types of data versioning tools

When choosing a data versioning tool, it's essential to think about how the tool works under the hood. Ignoring this aspect could lead to problems later, like limited storage space and increased latency.



These tools transfer the concepts of Git version control from code to data. They take snapshots of your data whenever there are changes you want to keep. These snapshots are called commits and store a list of all the saved changes.

These tools create data versions per hop. A hop is a “layer” of the data such as raw datasets, intermediate datasets or final datasets. Versions of data are stored inside a hub and only the delta of the changes are saved.

New data versioning tools are released frequently. It is likely that more architectures will develop over time.

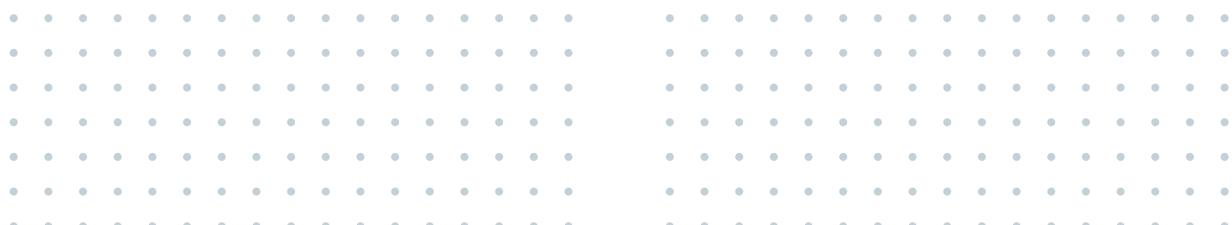
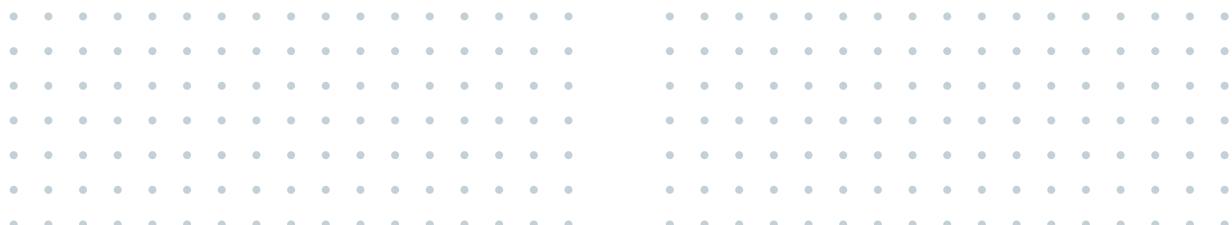
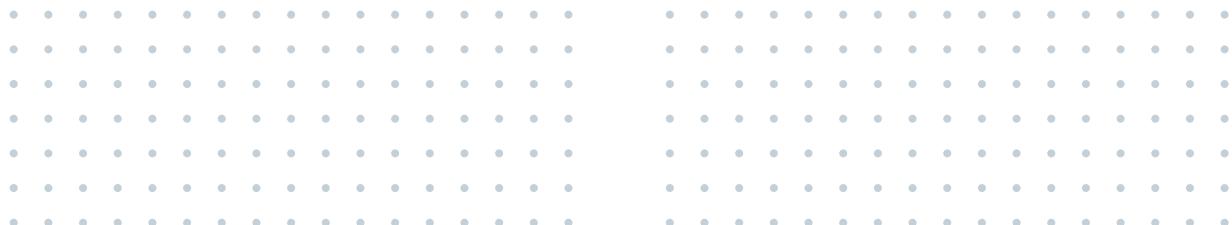
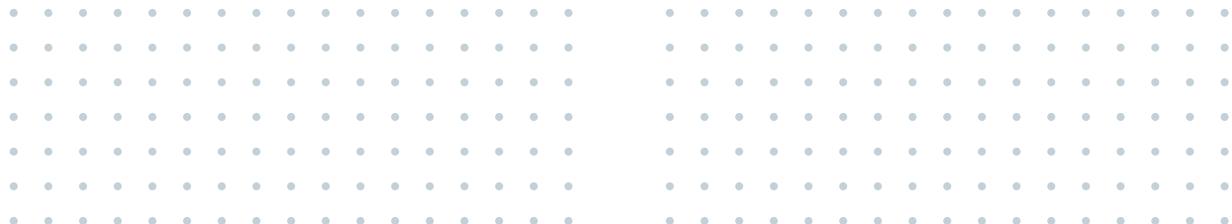
Numerous ways to evaluate data quality

Using data versioning tools alone won't solve data quality issues. Data quality refers to the reliability and accuracy of the data used to train and test models. It guarantees that the data is clean and error-free, leading to better and more trustworthy predictions. With this definition, try to name a few ways to verify data quality.

Validate that numerical values fall within predefined ranges.

Make sure that specific columns should not have any null (missing) values.

Based on these two answers, try to come up with your own

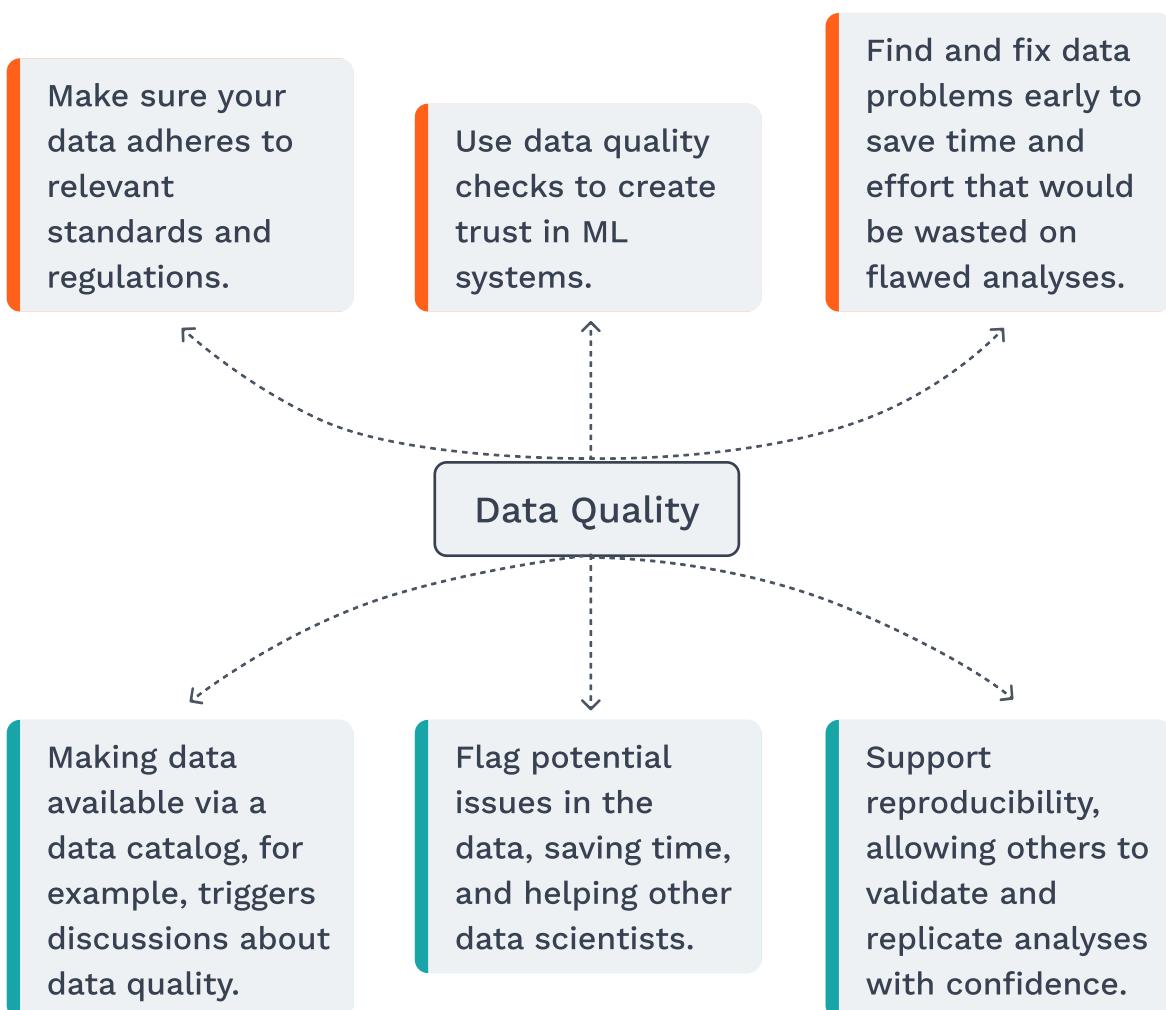


The two perspectives on data quality • Using and providing data

Poor data quality directly impacts the quality of your models. It's really important to have good data quality, not just for data scientists, but also if your team shares the data with others in the organization.

The Data User Perspective

Preparing and managing data for a specific use case

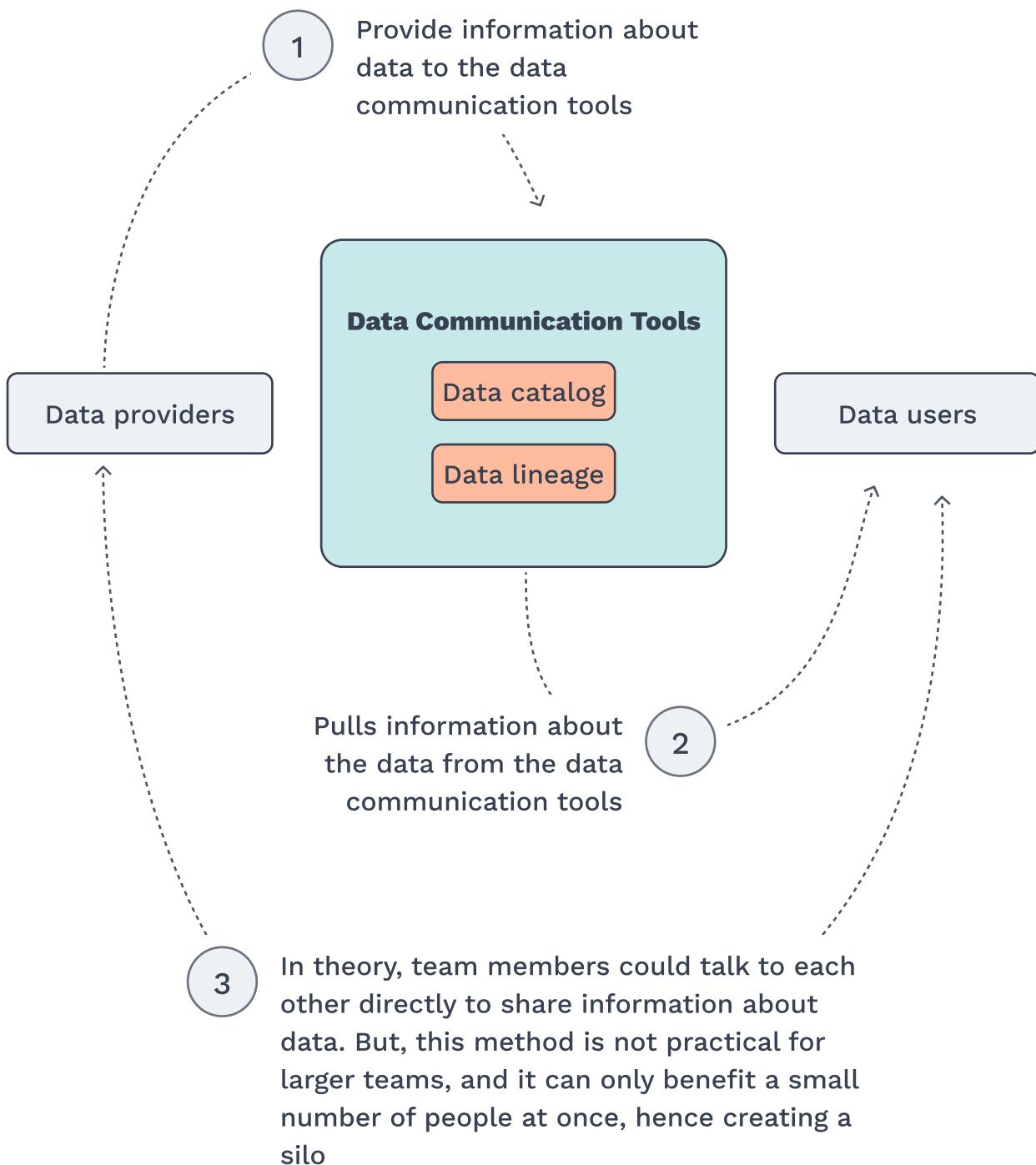


The Data Provider Perspective

Sharing the prepared data within the organization

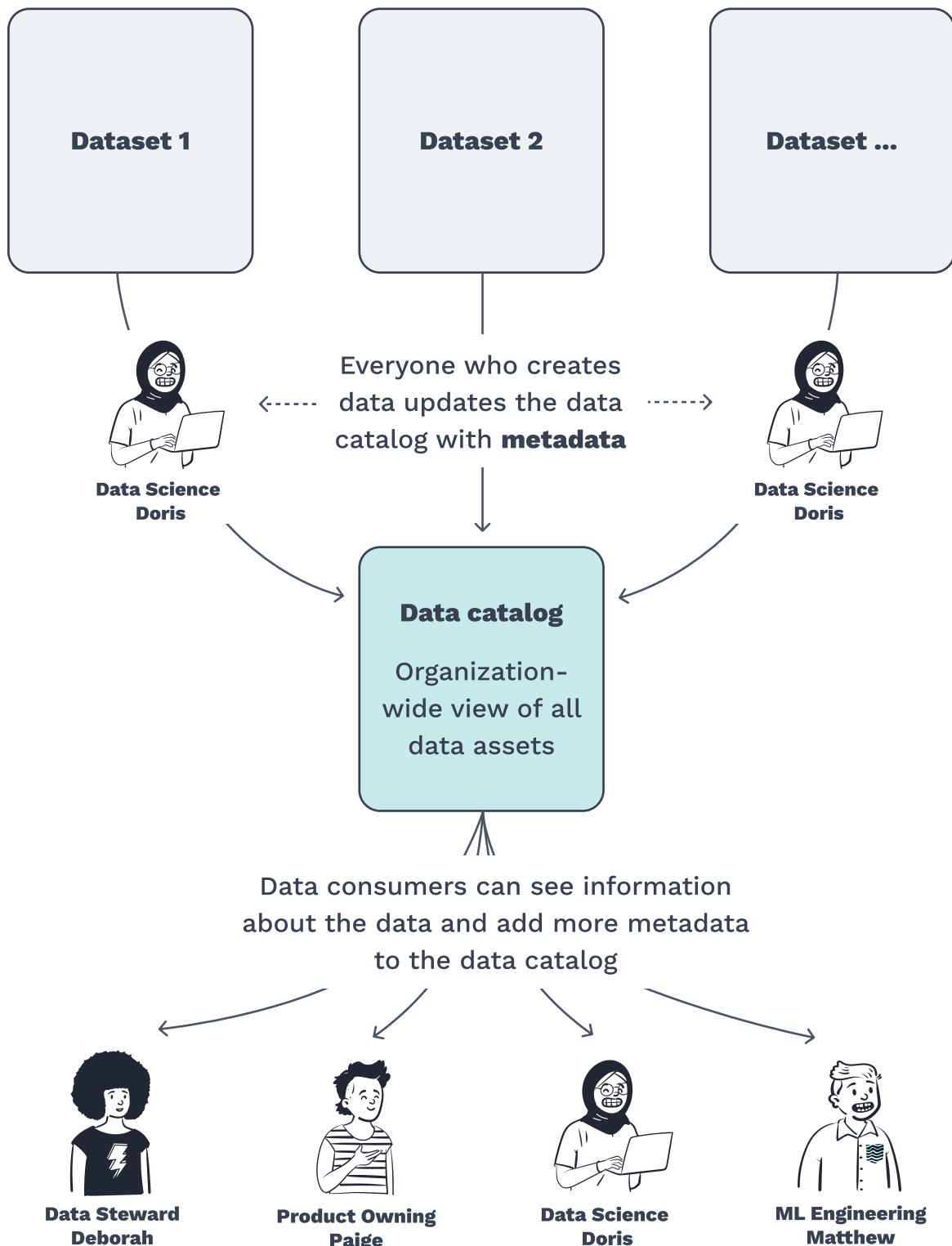
The benefits of data communication tools

As your team prepares the data and uses it for ML models, communication becomes increasingly important in the data lifecycle. You have to share information about your data with data users, preferably throughout the organization. The most effective way to achieve this is by using data communication tools.



Seamless communication about data through a data catalog

A data catalog is like a one-stop hub that shows you all your data, no matter where it's stored. It helps you see and find your data more easily within your organization, making it simpler to discover and manage your data effectively.



What a data catalog provides and doesn't provide

It's essential to define the boundaries of data catalogs and understand what they can and cannot offer. Try to describe these boundaries.

A data catalog provides ...

- A uniform grid of small, light gray dots arranged in a regular pattern across the entire frame.

A data catalog does not provide ...

- A uniform grid of small, light gray dots arranged in a regular pattern across the entire page.

Benefits of a data catalog from the perspective of the data steward

A data catalog is beneficial for data governance, which involves managing and controlling an organization's data. Data stewards use a data catalog to answer important questions about the organization's data. Try to name some of these questions that data stewards can answer with data catalogs.



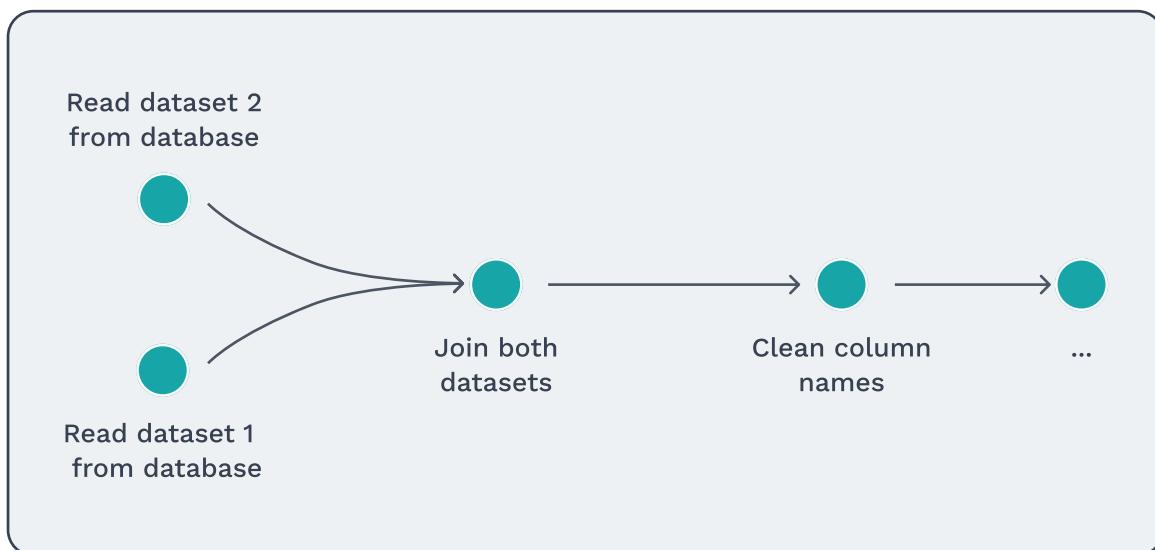
Data Steward
Deborah



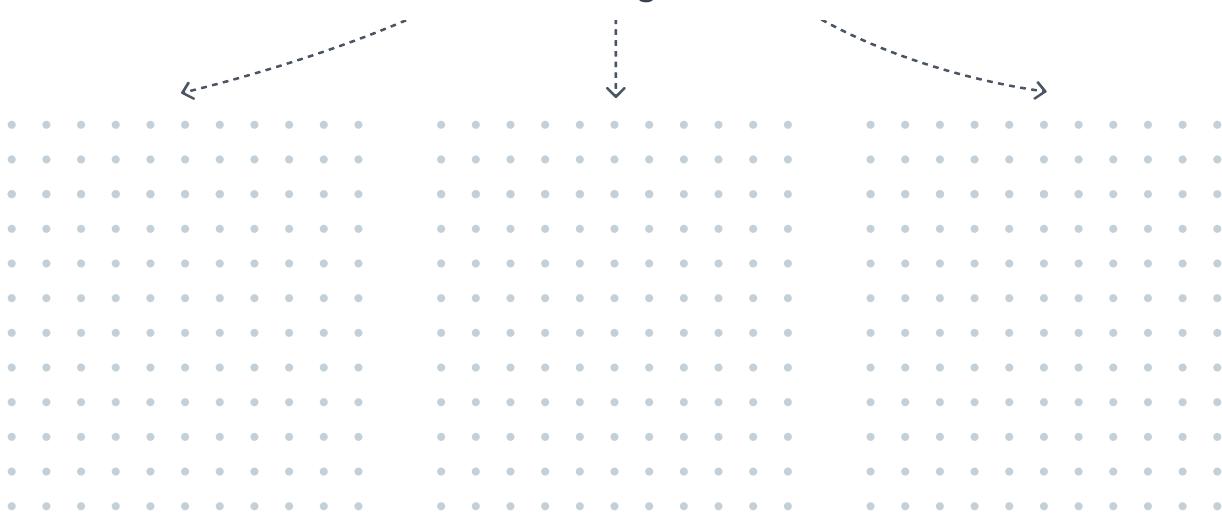
Inspecting data preparation • Data lineage

Data lineage is a way to track and trace the data's journey. It shows where the data originated, how it was transformed along the way, and where it ends up being used. This comprehensive view allows you to understand the data's journey from start to finish. Try to describe which accountabilities profit from a data lineage view and why.

A simplified data lineage view



How do accountabilities profit from looking
into the lineage of data?



Practical recommendations

If you're uncertain about what to do next, consider these practical recommendations.

Keep in mind that there are different tools for versioning data, and each one works best for specific situations. When choosing a data versioning system, consider the features of these tools and their suitability for your use case.

Data catalogs and data lineage are tools for communication. They help solve the problem of sharing information about the ML system with others. But for them to work well, your team needs to contribute the metadata of your data to the data catalog.

Make sure to talk openly with your team and company about your data architectures and how you process data (ETL/ELT patterns). Doing this will help you build ML systems more effectively while also following the ML Principles.

Self-assessment of comprehension

Take a moment to answer the following questions about the content covered in this module. Keep in mind that there's only one correct answer for each question.

1. What is the difference between data lineage and data versioning?

- A) Data lineage provides a Git-like functionality for creating multiple branches of data for experiments, while data versioning creates snapshots of data over time.
- B) Data lineage tracks the flow and transformations of data, while data versioning tools are packaging tools that version the data being shipped to the user.
- C) Data lineage tracks the flow of data, while data versioning manages different versions of data.

2. Which of the following analogies best describes the function of a data catalog?

- A) A data catalog is like a cooking recipe book, providing step-by-step instructions for data analysis.
- B) A data catalog is like a library catalog, organizing and categorizing data assets for easy discovery and retrieval.
- C) A data catalog is like a weather forecast, predicting future trends and patterns in data usage.

3. How does the Git-Pattern architecture for data versioning differ from the Multi-Hop Architecture?

- A) Multi-Hop architectures version data per step in the data pipeline (e.g. raw data, intermediate data). Git-Pattern architectures, however, version data on the directory and file level with commits inside branches.
- B) Multi-Hop architectures use databases as their foundation, whereas Git-Pattern architectures are built on Git repositories.
- C) Multi-Hop architectures do two things at once: they keep track of data lineage and also handle data versioning. In contrast, Git-Pattern architectures are specifically designed for data versioning, focusing solely on that aspect.

Self-assessment of comprehension

4. How do data catalogs and data lineage help teams communicate about data in different ways?

- A) Data catalogs store data in a centralized repository that anyone can access, while data lineage describes how the data got into the repository.
- B) Data catalogs provide a method for indirectly accessing data outside your team, while data lineage allows you to see who downloaded the data.
- C) Data catalogs provide metadata that describes the various datasets in an organization, while data lineage describes where the data originated and how it was changed.

5. Which of the following scenarios is best suited for an ELT pattern?

- A) When the columns of your data won't change
- B) When you need to reproduce your ML models
- C) When your team needs to generate up-to-date reports and analysis using standard benchmarks on relational data.

6. Which of the following is a feature of a Data Lake?

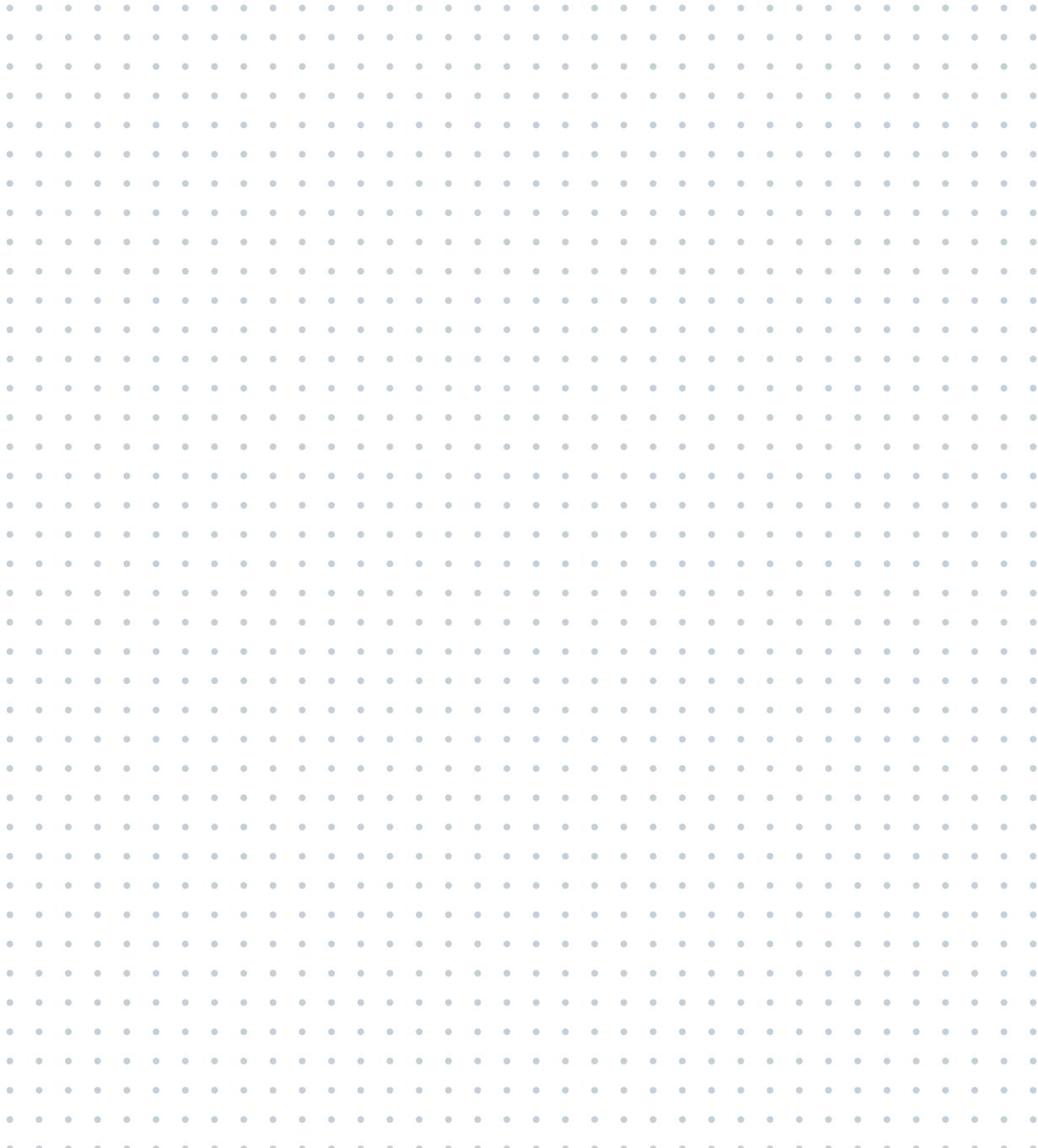
- A) It follows an ETL pattern
- B) It stores structured, unstructured and semi-structured data
- C) It only stores structured data

7. Which of the following features distinguishes a Data Lakehouse from a Data Lake?

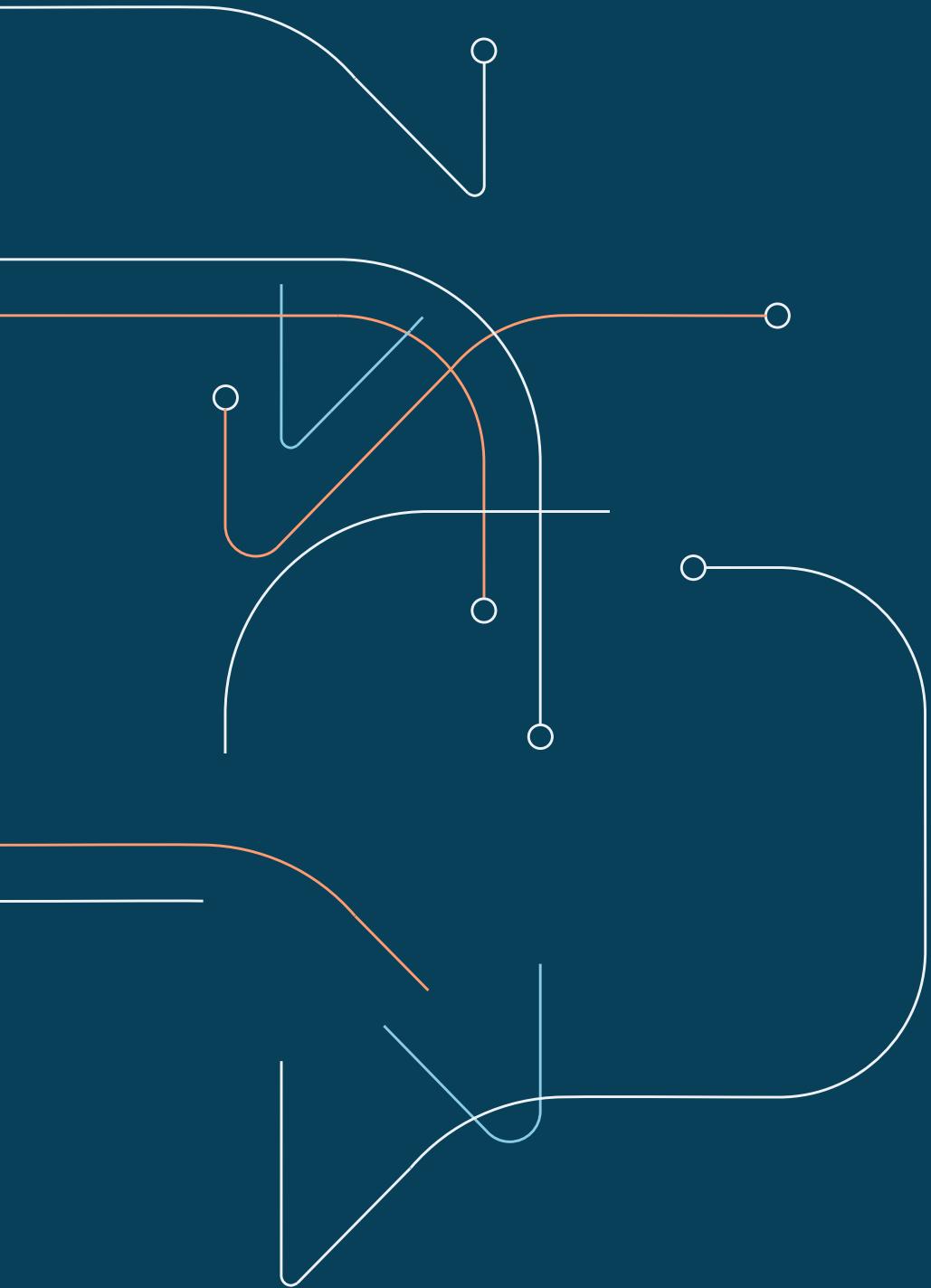
- A) Support of data queries via structured query language
- B) Storage of raw, untransformed data
- C) It allows for ELT

Module review

Each module begins and ends with this page. At the beginning of the module, we asked you to write down some questions that you would like answered at the end. Once you have written down these questions and worked through the content, take some time to look at the questions again and explain the answers to yourself.

A large grid of grey dots on a white background, intended for handwritten notes. The grid consists of approximately 20 columns and 30 rows of dots.

04 Modeling



Intended learning objectives

In this module, our main focus lies in improving Model Management. We will delve into the components of machine learning experiments, highlighting the differentiation between model versioning and experiment tracking.

Furthermore, we'll explore how a model registry can facilitate the promotion of your models to production. Additionally, we'll discuss how model registries and model versioning support collaboration, and provide insights into structuring model management within your project setup.

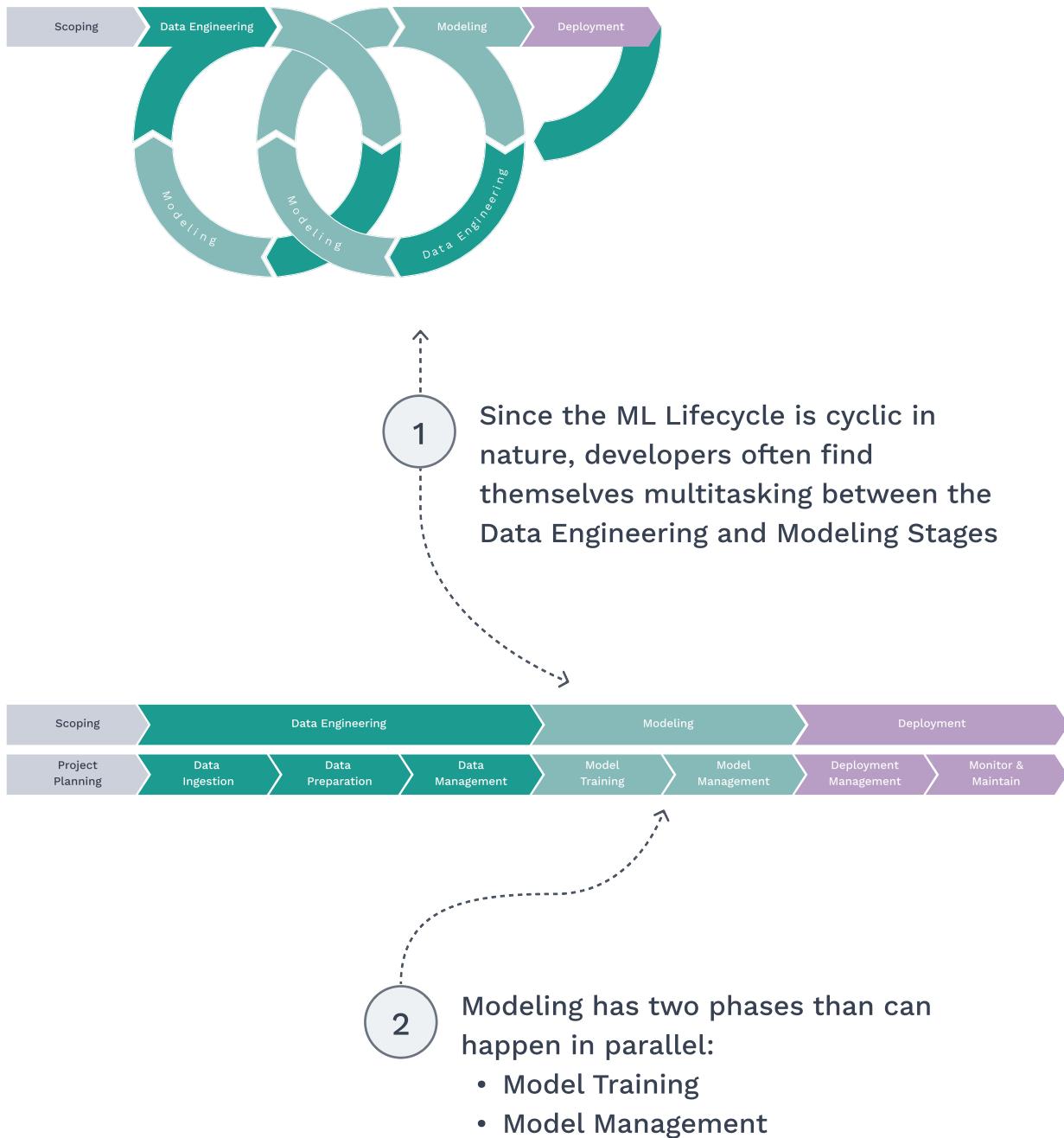
By the end of this module, you will have developed the following proficiencies:

- ✓ Name the key components of an ML experiment.
- ✓ Differentiate between model versioning and experiment tracking.
- ✓ Identify and name the different areas of a model registry.
- ✓ Explain the collaboration between data scientists and ML engineers during the Modeling Stage.
- ✓ Describe the need for auto-scaling in the context of training ML models.
- ✓ Explain the process by which your team could determine when to prioritize Model Management throughout the project.

Before you begin, we recommend that you take a look at the contents of the module. Take a few minutes to go through each page and look at the headings and visuals. Try to get an overview of the module's content. When you are done, go to the last page of the module and write down a few questions you would like to have answered at the end of the module. We will ask you to review these questions later after you have worked through the content.

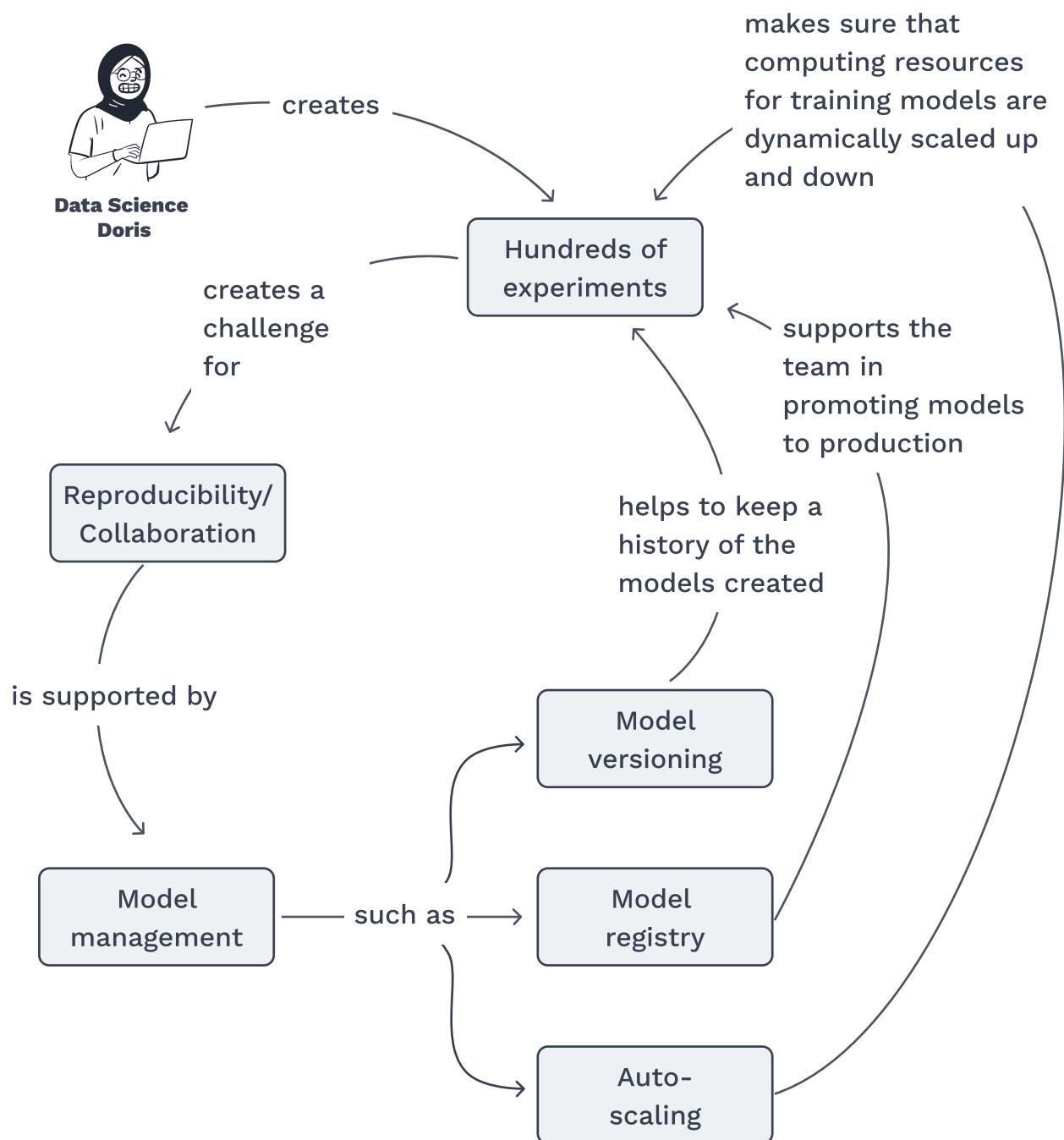
Introduction to the Modeling Stage

The main aim of modeling is to build a machine learning model that fulfills the system requirements. This stage includes a Model Management Phase, which promotes smooth collaboration among team members and makes sure the models can be reproduced and inspected.



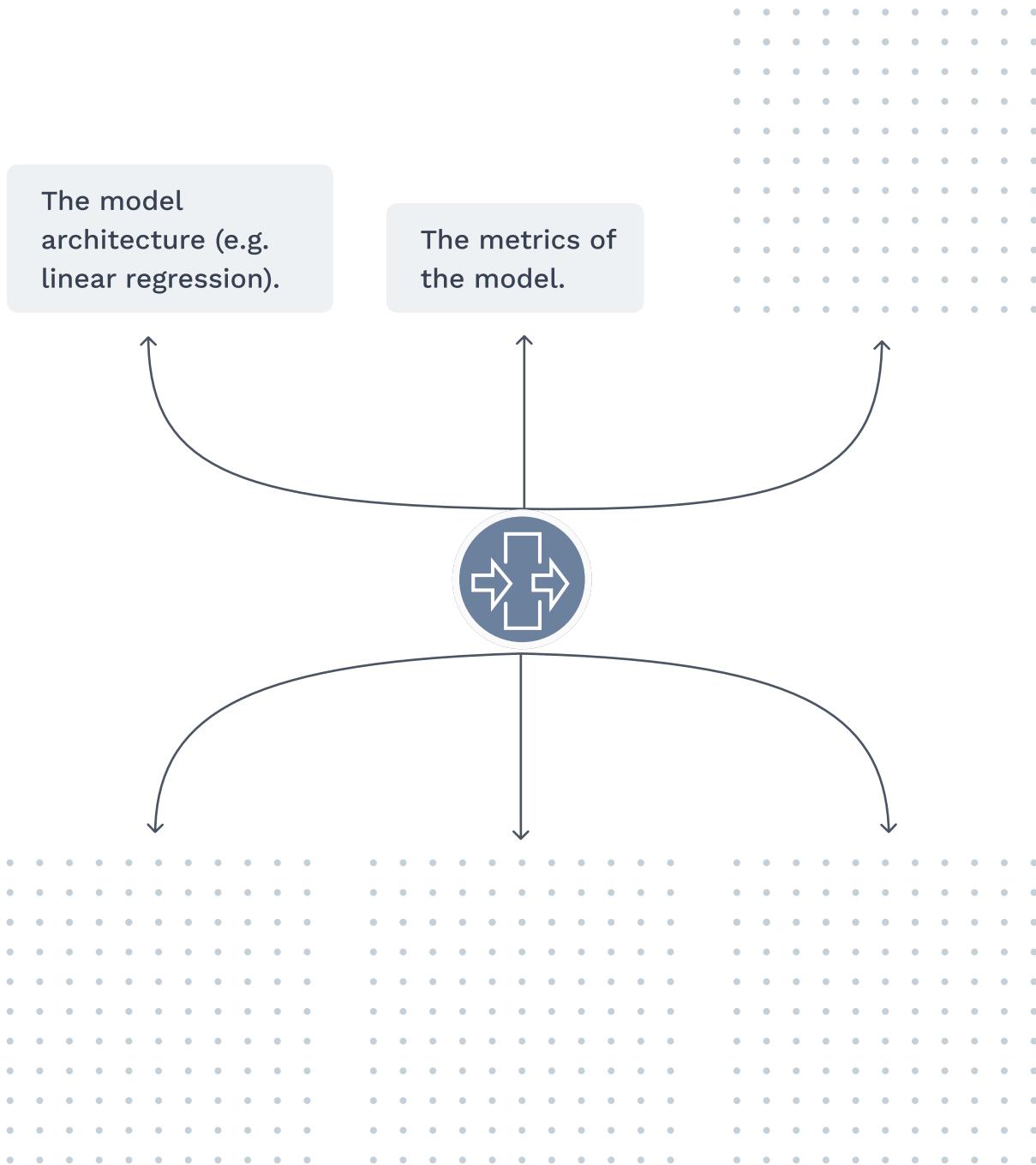
The lifecycle of models

The biggest challenge in modeling is managing and making sense of the many models that are created at that stage. The following concept map outlines how Model Management can help you in this stage.



Components of an ML experiment

The term experiment often goes together with the concept of modeling. An experiment includes both the inputs and the outputs of a machine learning model. Your task is to find the parts of an experiment. We've already given you two to start with.



Experiment tracking and model versioning

There are two methods to handle experiments: model versioning and experiment tracking. Even though the terms experiment tracking and model versioning are often used interchangeably, they have different meanings.

Experiment tracking

Refers to process of recording information for an individual experiment. The output is a model version.

	Time	Duration	Source	Metrics	...
1	2023-05-02	10 min	Users/aai...	rmse = 0.76	
2	2023-04-30	12 min	Users/aai...	rmse = 0.72	
3	2023-04-27	9 min	Users/aai...	rmse = 0.78	
...

Model versioning

Refers to the process of maintaining a history of models over time.

Questions you can answer with experiment tracking and model versioning

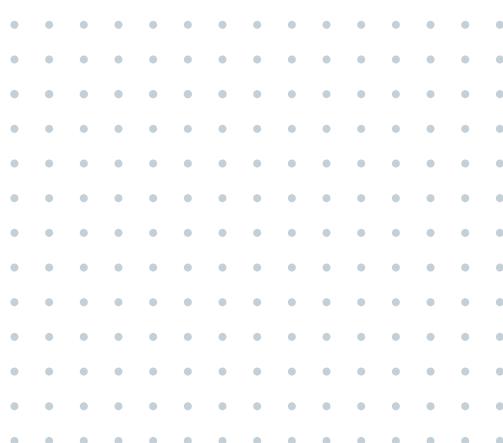
With experiment tracking and model versioning your team can answer a heap of questions more efficiently. Try to name one question for each accountability that can be answered with experiment tracking and model versioning.



Data Science
Doris



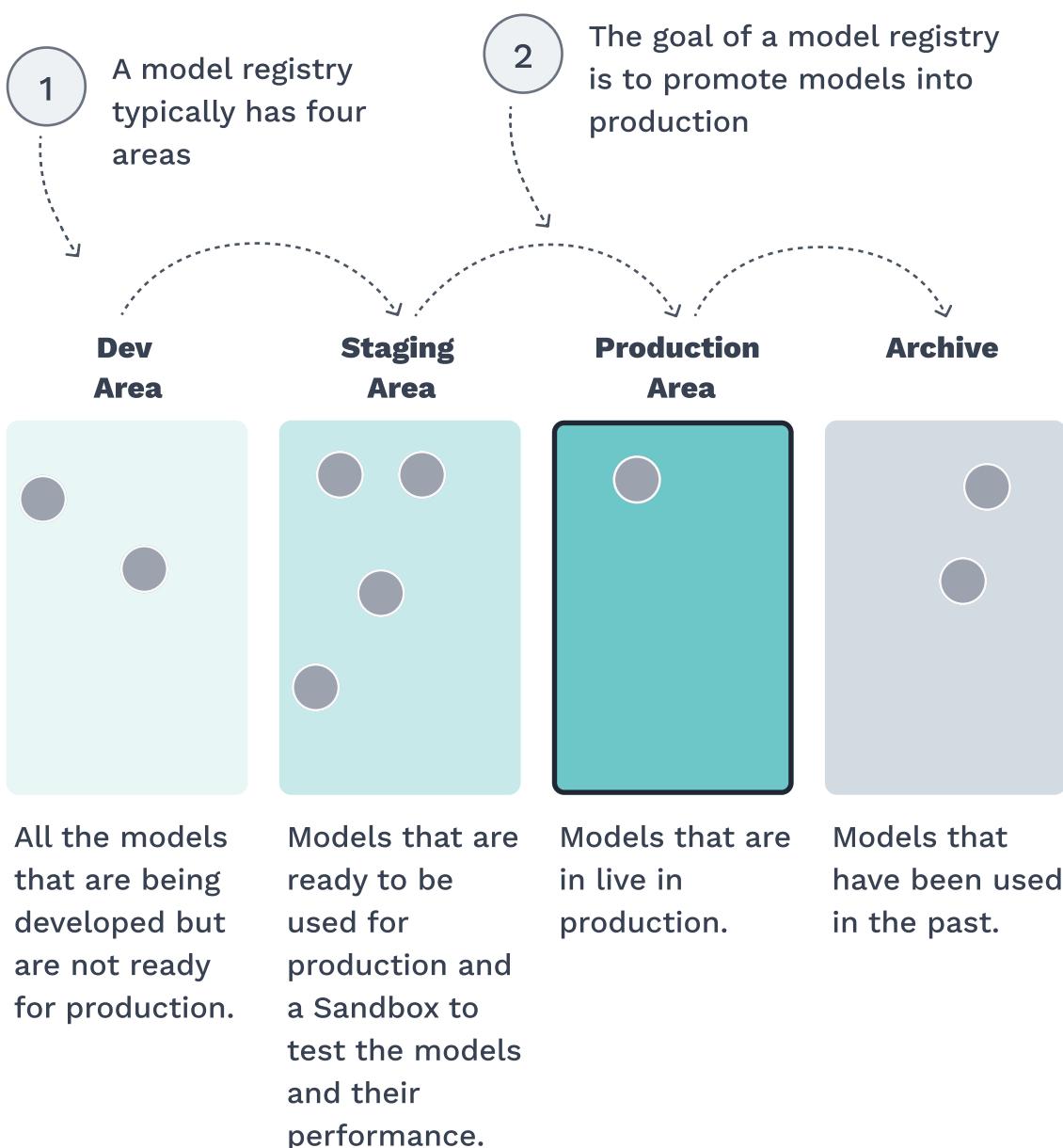
Data Steward
Deborah



Product Owning
Paige

A model registry supports promoting models to production

The goal of a model registry is to seamlessly enable the promotion and governance of models for the ML team. The registry is a store where all the models and their versions can be stored.

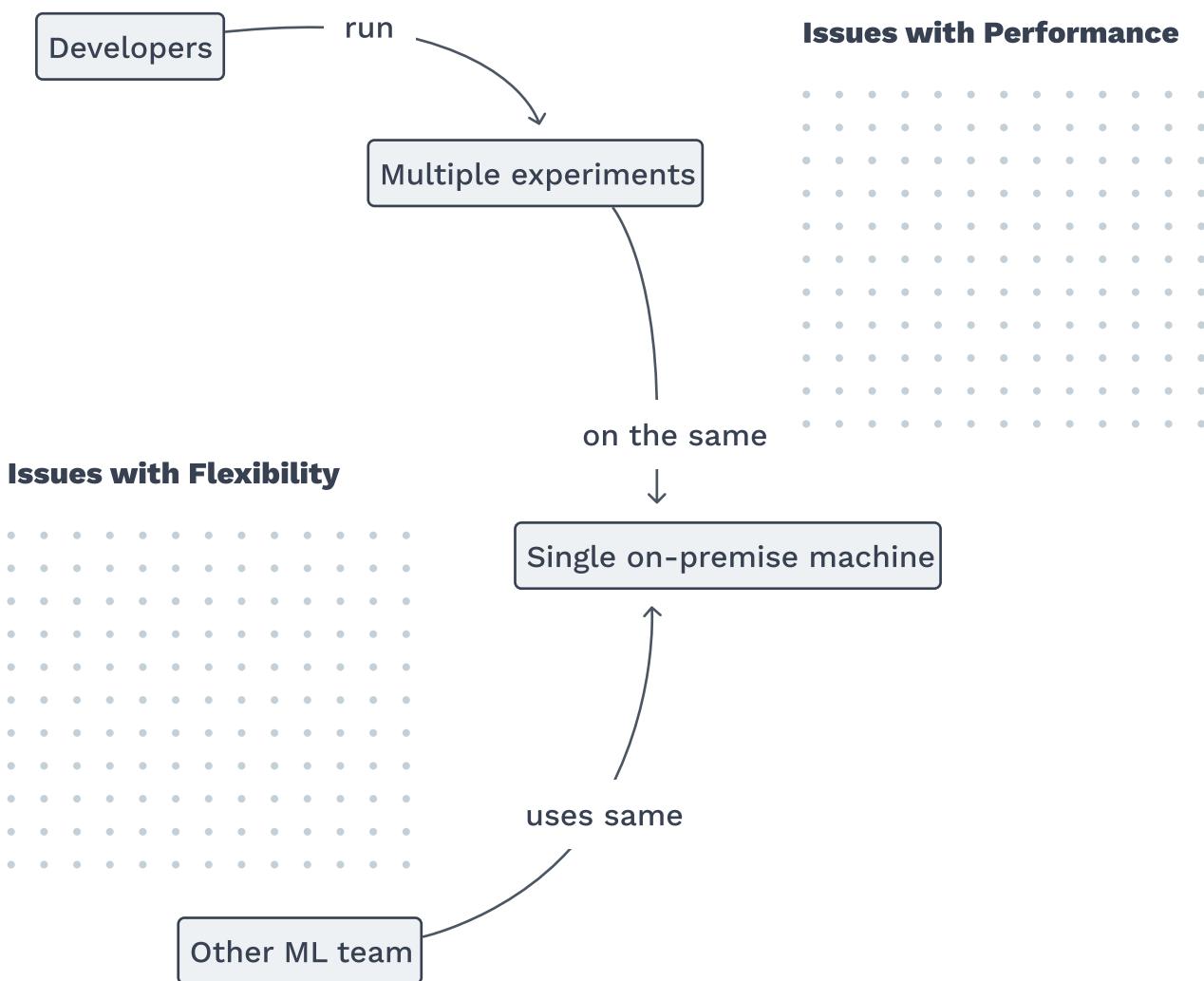


Resources

<https://neptune.ai/blog/model-registry-makes-mlops-work>

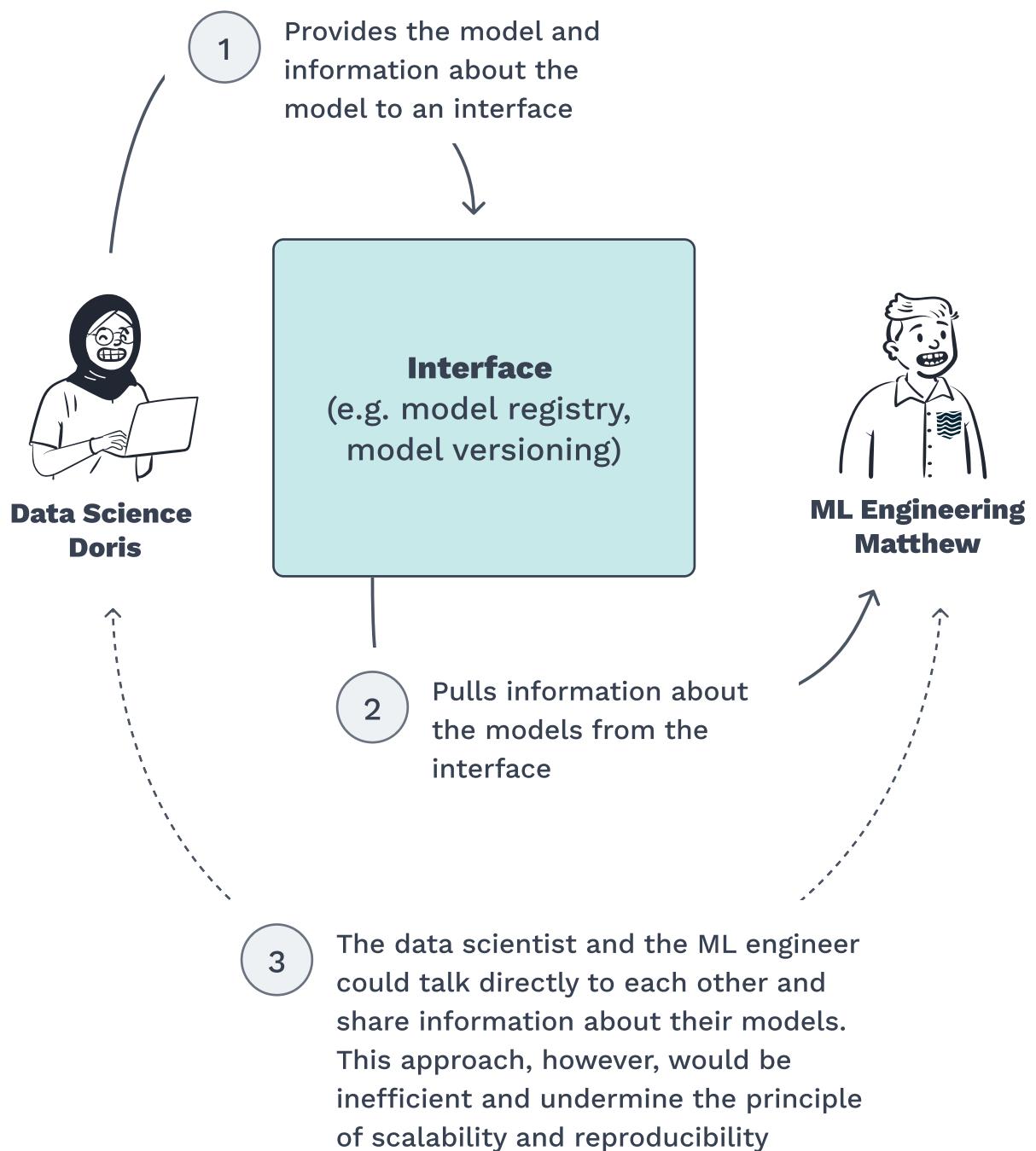
The need for auto-scaling

Auto-scaling allows ML systems to adapt to varying workloads and handle increased demand effectively. By automatically scaling up or down the available computing resources, ML systems can maintain optimal performance and efficiency. Here's an example of what could happen if teams don't use Auto Scaling. Try to describe the issues with performance and flexibility that come up in this situation.



In model management collaboration is facilitated through an interface

In the Modeling Stage, Data Science and ML Engineering are the main accountabilities. In smaller teams, one person often performs both accountabilities. Model Management is essentially an attempt to create an interface through which both accountabilities can exchange information about their models.



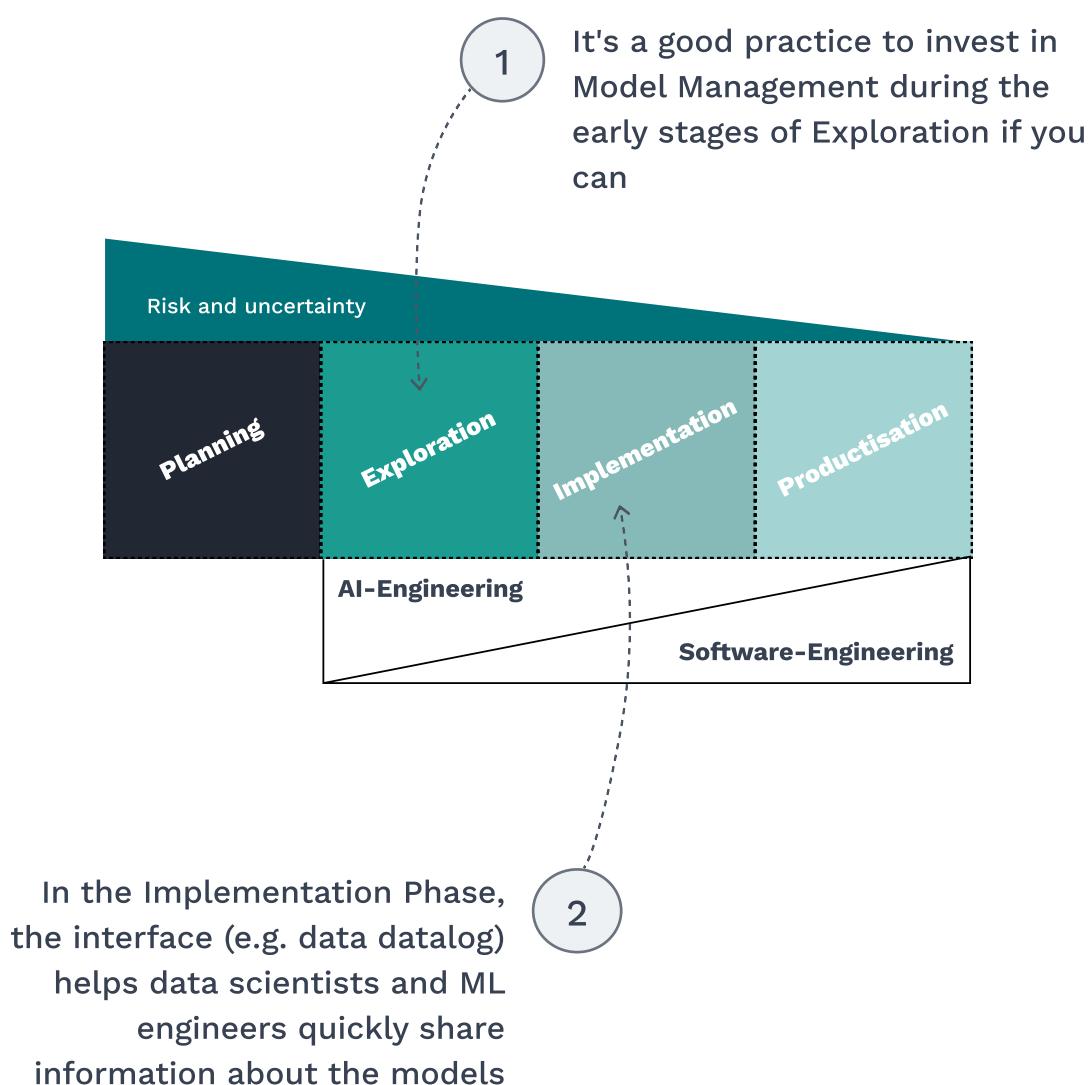
How data scientists can collaborate with ML engineers through the interface

Data scientists are responsible not only for versioning the models, but also for adding relevant information about the models to the model versioning system. Try to find out what information data scientists can add to the interface to help ML engineers perform their tasks effectively.



Model management becomes more important as the project progresses

As the project advances, an increasing amount of time is dedicated to Model Management. Additionally, there emerges a growing necessity to document information about the experiments and establish effective processes for serving models.

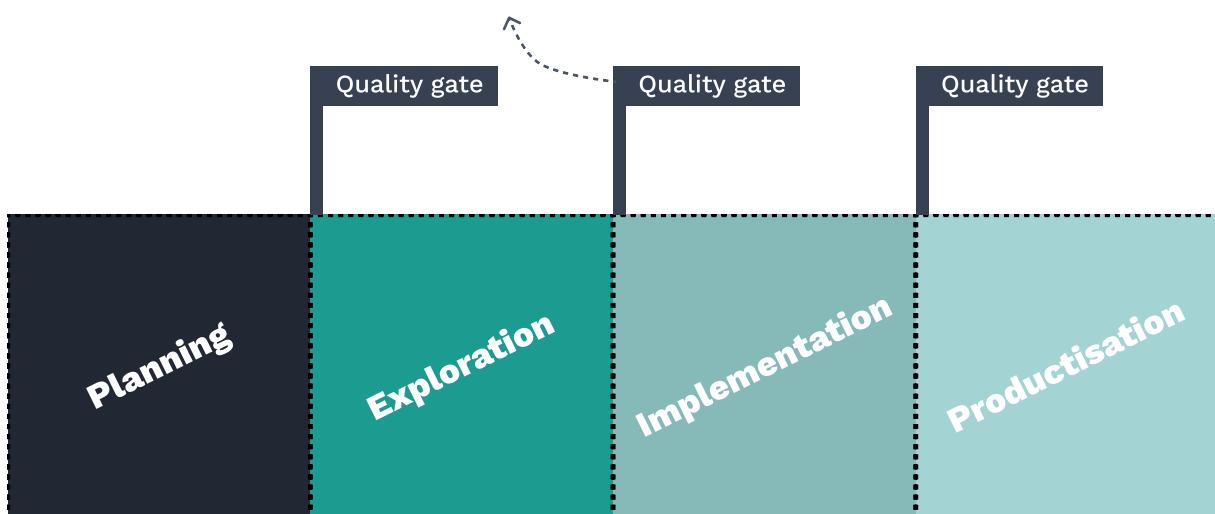
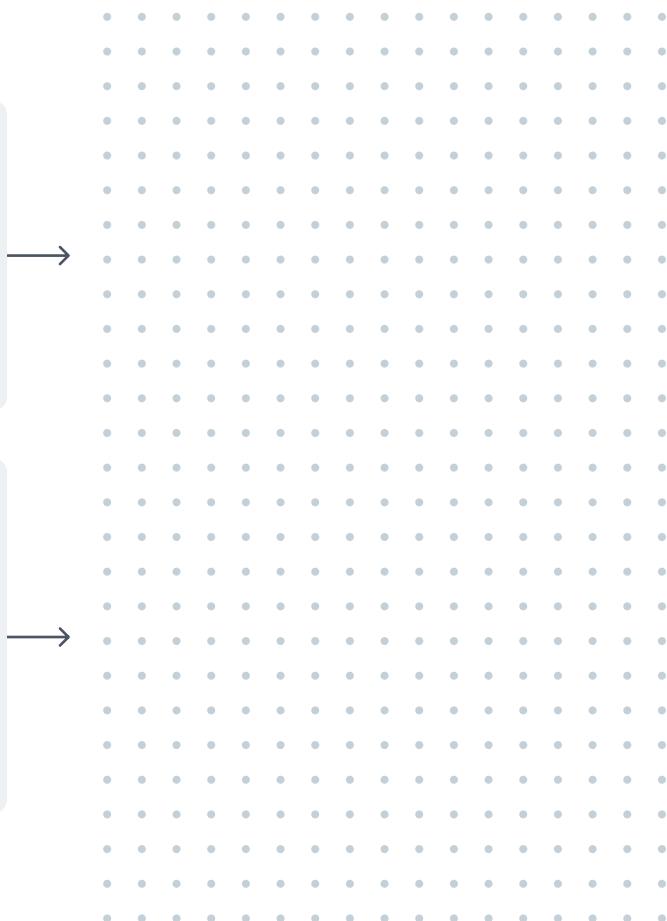


How model management supports the deliverables of the Exploration Phase

An important part of Model Management is keeping a record of information about the models that have been trained. This information is crucial for the quality gate at the end of the Exploration and Implementation Phase. Here you can see two quality gates for the Exploration Phase. Try to describe how the Model Management supports these quality gates.

We have identified several promising ways to achieve our optimization and model-dependent goals. These paths adhere to all technical requirements.

We have a functional development setup that fosters smooth communication and collaboration between developers and technical stakeholders.



Practical recommendations

If you're uncertain about what to do next, consider these practical recommendations.

With your team, choose when it's best to concentrate on Model Management during the project. In some cases, you can set up basic Model Management during the Exploration Phase, while for other use cases, the focus on Model Management should be during the Implementation Phase.

Model Management is not just about technology. If your data scientists aren't giving the model version system the right information and aren't clearly promoting models in the model registry, it will impact the efficiency of Model Management. Make sure your team is aware of this.

Setting up auto-scaling is quite fast since major cloud providers offer this feature. Talk to your IT team about how your computing needs will change during the project, and decide if auto-scaling is necessary.

Self-assessment of comprehension

Take a moment to answer the following questions about the content covered in this module. Keep in mind that there's only one correct answer for each question.

1. What is the difference between experiment tracking and model versioning?

- A) Experiment tracking involves keeping records on the experiments simultaneously, while model versioning focuses on maintaining a record of changes made to a model over time.
- B) Experiment tracking refers to the process of monitoring the progress of a single experiment and recording its outcomes and conclusions, while model versioning is the practice of organizing and managing datasets within a single experiment.
- C) Experiment tracking describes the process of tracking the information of a single experiment, whereas model versioning is the process of tracking multiple experiments over time.

2. During the handover process in the Modeling Stage, which two accountabilities are crucial?

- A) Data scientist and ML engineer
- B) Data engineer and ML engineer
- C) Data scientist and solution architect

3. Name the typical four areas of a model registry

- A) Development, Production, Archive, Staging
- B) Testing, Production, Archive, Staging
- C) Development, Local, Archive, Staging

Self-assessment of comprehension

4. Which of the following components is not part of an experiment in the Modeling Stage?

- A) The hyperparameters
- B) The output logs
- C) The parquet data file

5. Your product owner wants to have a look at the models that are currently in production at any given time. Which of the methods discussed would solve this problem?

- A) Experiment tracking
- B) Auto-scaling
- C) Model registry

6. What is the key problem that is being solved through auto-scaling?

- A) The system provides the ability to allocate additional resources based on the workload, providing flexibility for scaling up or down resources in response to demand fluctuations.
- B) The system automatically organizes tasks among multiple team members and maximizes task completion to ensure efficient model training.
- C) The system enables developers to train multiple models concurrently, evaluating them against one another to identify the model with the most favorable metrics.

Self-assessment of comprehension

7. What is the most important quality gate that needs to be considered in terms of modeling to move from Implementation to Production?

- A) The model performance fulfils the requirements.
- B) An infrastructure is in place that is capable of continued iterations.
- C) The team has a well-defined list of features to implement, along with estimates or validations of their optimization potential.

8. Which of the following analogies best describes the function of a model registry?

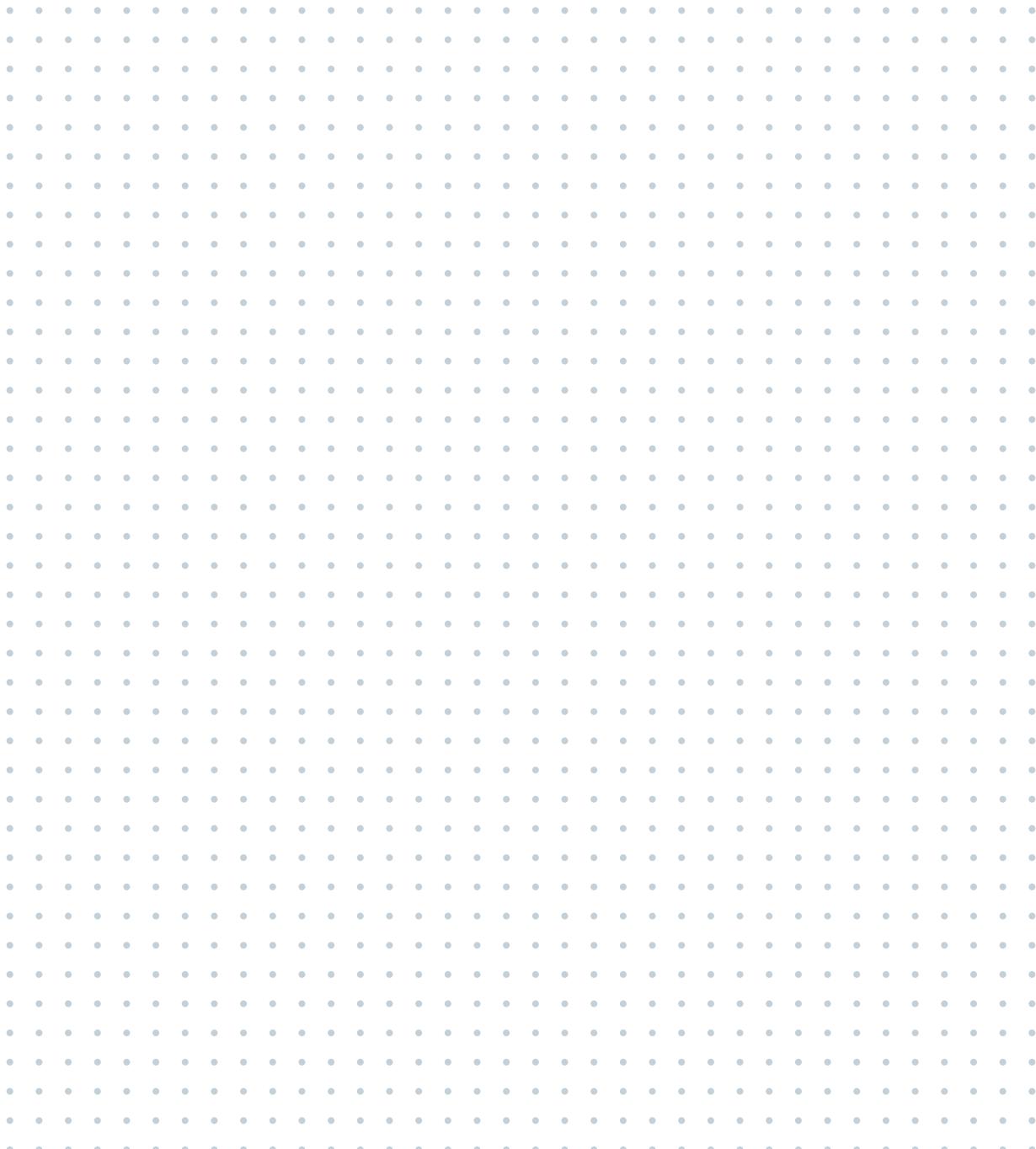
- A) A model registry is like a coat rack for storing models.
- B) A model registry is like a playground for testing models.
- C) A model registry is like version control, but for machine learning models.

9. What is the main benefit of a model registry for data scientists and ML engineers?

- A) One of the key benefits of having a model registry for data scientists and ML engineers is that it provides them with virtually unlimited storage capacity for their models. This ensures that they never have to worry about running out of space, even when dealing with large and complex models.
- B) The model registry offers a platform for data scientists and ML engineers to create experiments, simplifying the process of iterating on model building.
- C) A model registry facilitates collaboration among data scientists and ML engineers, allowing them to store and manage models in a centralized repository. It promotes reproducibility by preserving model versions and tracking deployment state (staging, production, archive), enabling researchers to replicate experiments and compare results.

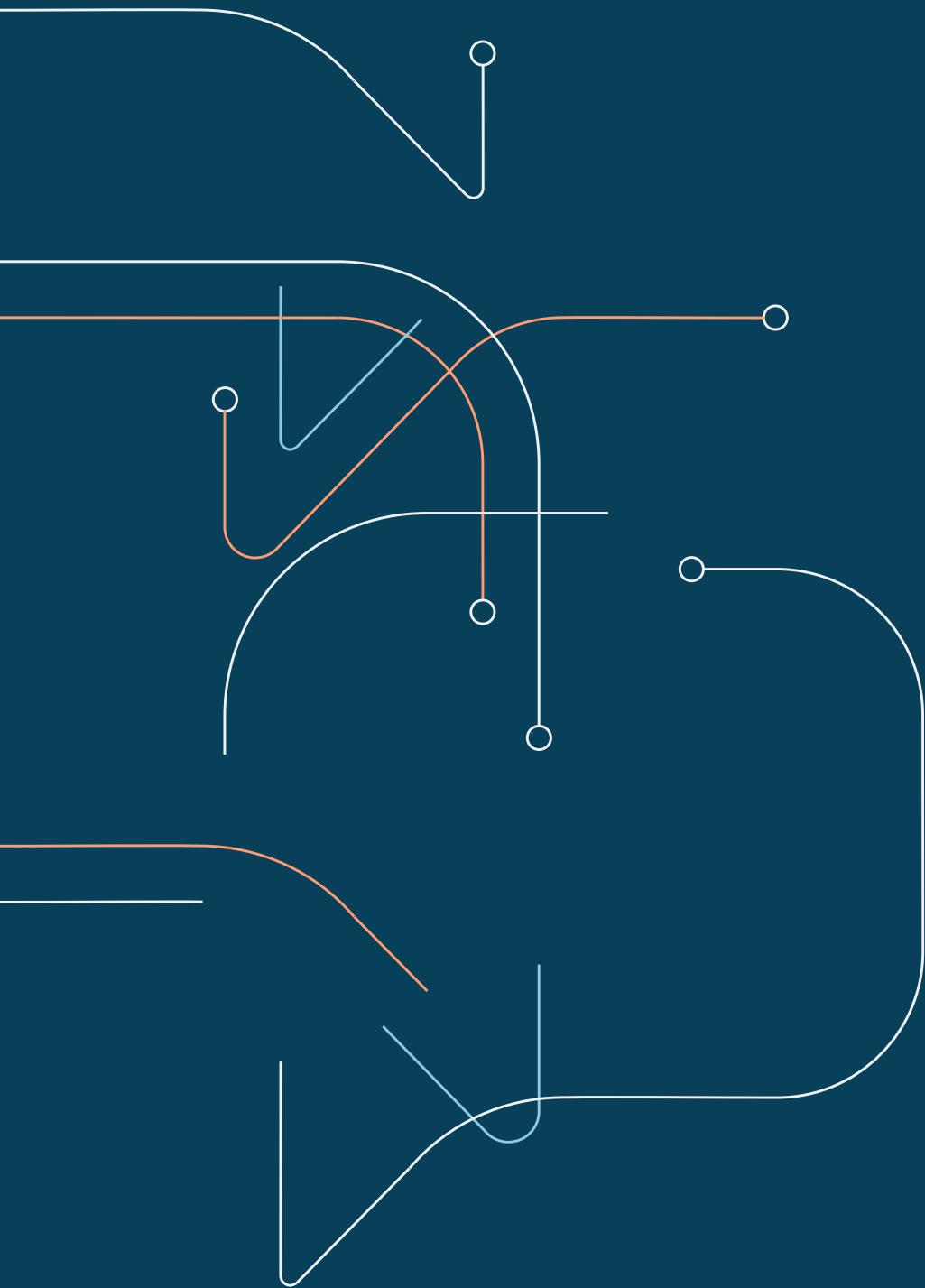
Module review

Each module begins and ends with this page. At the beginning of the module, we asked you to write down some questions that you would like answered at the end. Once you have written down these questions and worked through the content, take some time to look at the questions again and explain the answers to yourself.

A large grid of grey dots on a white background, intended for handwriting practice or writing responses.

0 5

Deployment



Intended learning objectives

This module is dedicated to improving your deployment practices. We will look at the various ways ML models can be deployed. We will also explore the many ways in which ML models can become obsolete after they have been put into production.

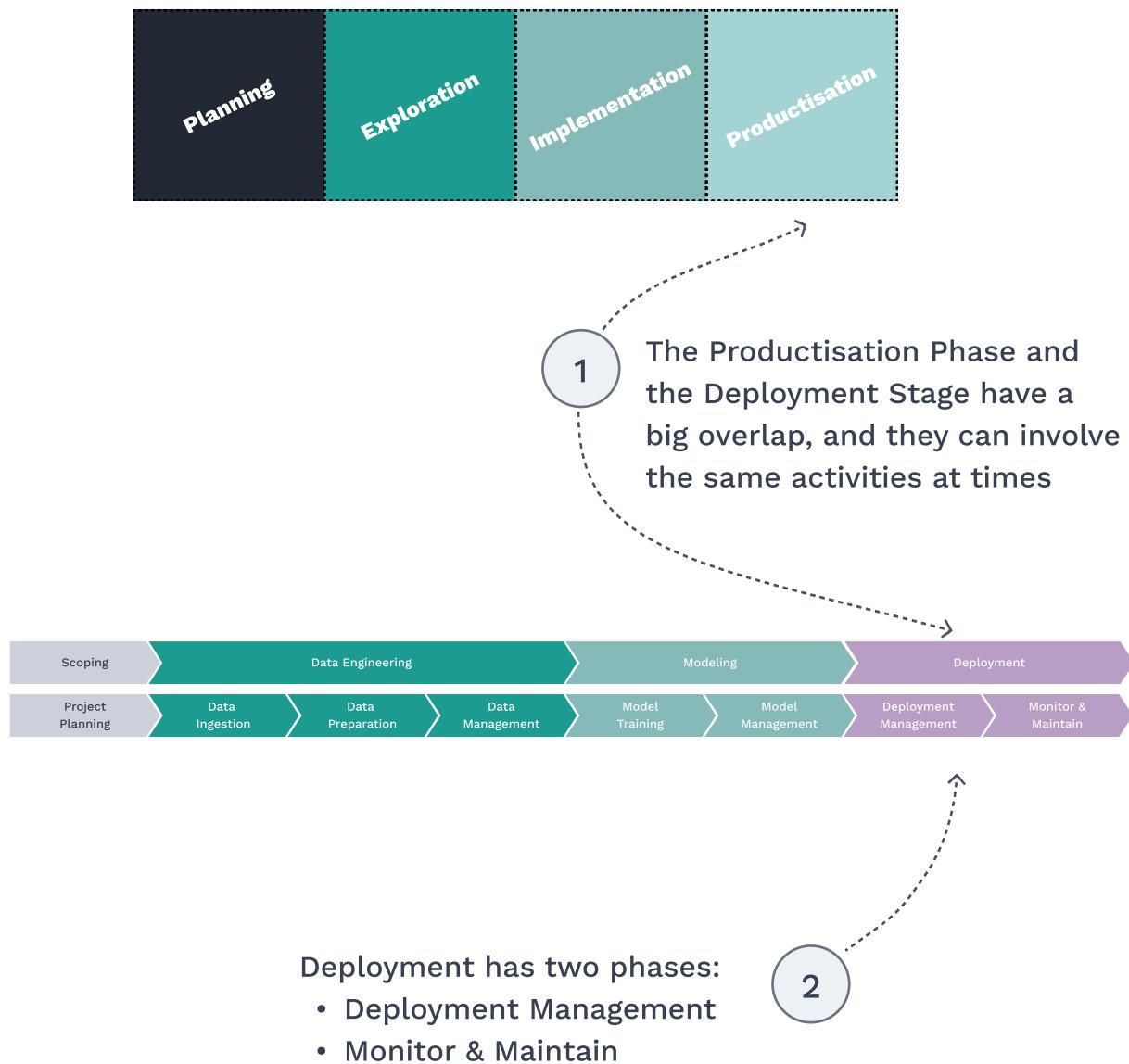
By the end of this module, you will have developed the following proficiencies:

- ✓ Explain the function of an inference pipeline.
- ✓ Explain three model serving patterns model-as-a-dependency, model-as-a-service, and precompute-service-pattern.
- ✓ Explain three deployment strategies A/B testing, multi-armed bandit, and shadow deployment.
- ✓ Select the most suitable deployment strategies and model service patterns for fictional scenarios.
- ✓ Explain the concepts of data drift, covariate shift, label shift, and concept drift; and the types of concept drift.
- ✓ Name at least two metrics used for system monitoring.

Before you begin, we recommend that you take a look at the contents of the module. Take a few minutes to go through each page and look at the headings and visuals. Try to get an overview of the module's content. When you are done, go to the last page of the module and write down a few questions you would like to have answered at the end of the module. We will ask you to review these questions later after you have worked through the content.

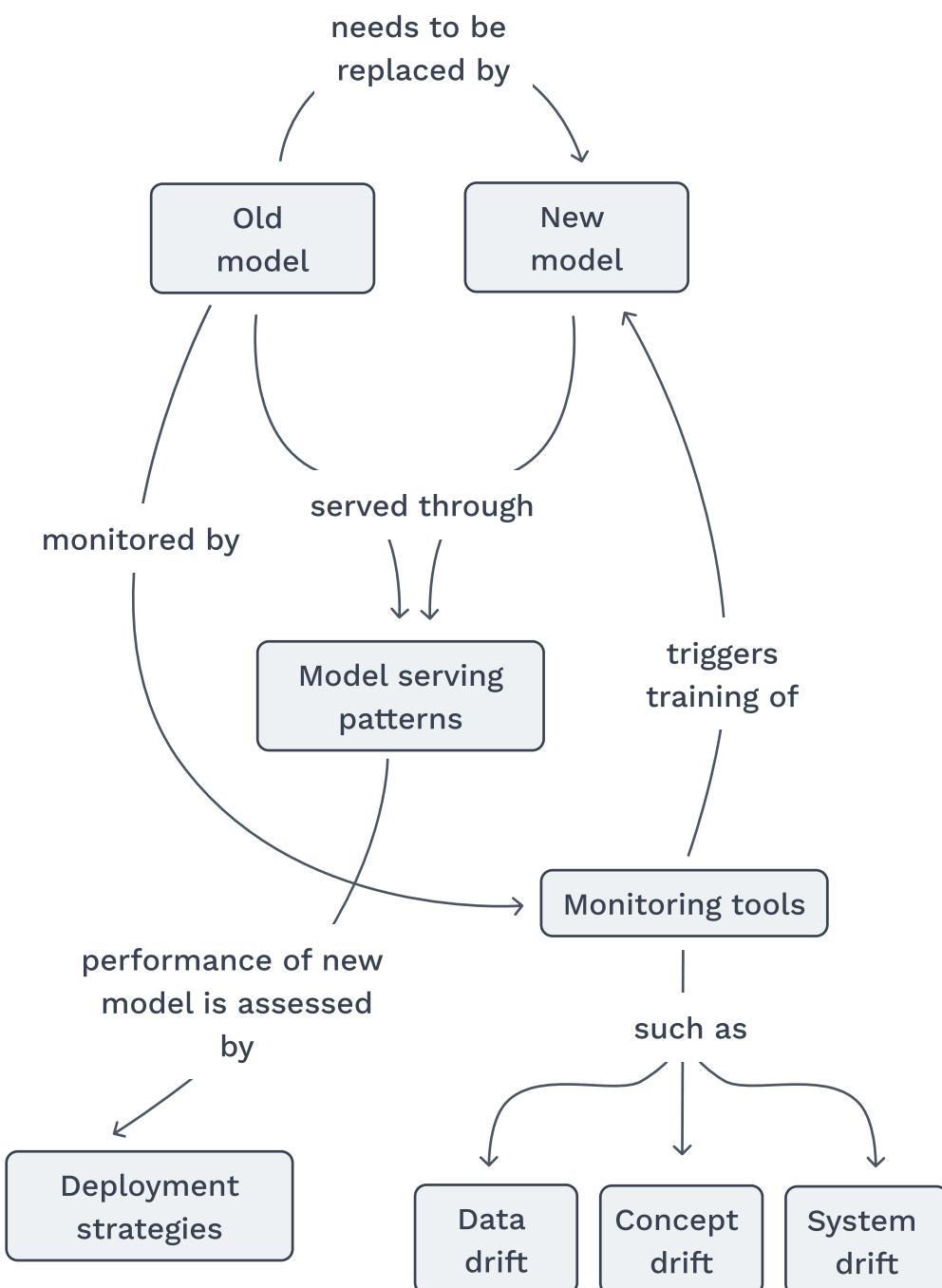
Introduction to the Deployment Stage

The goal of the Deployment Stage is to put machine learning models into production and to monitor their status in the production environment.



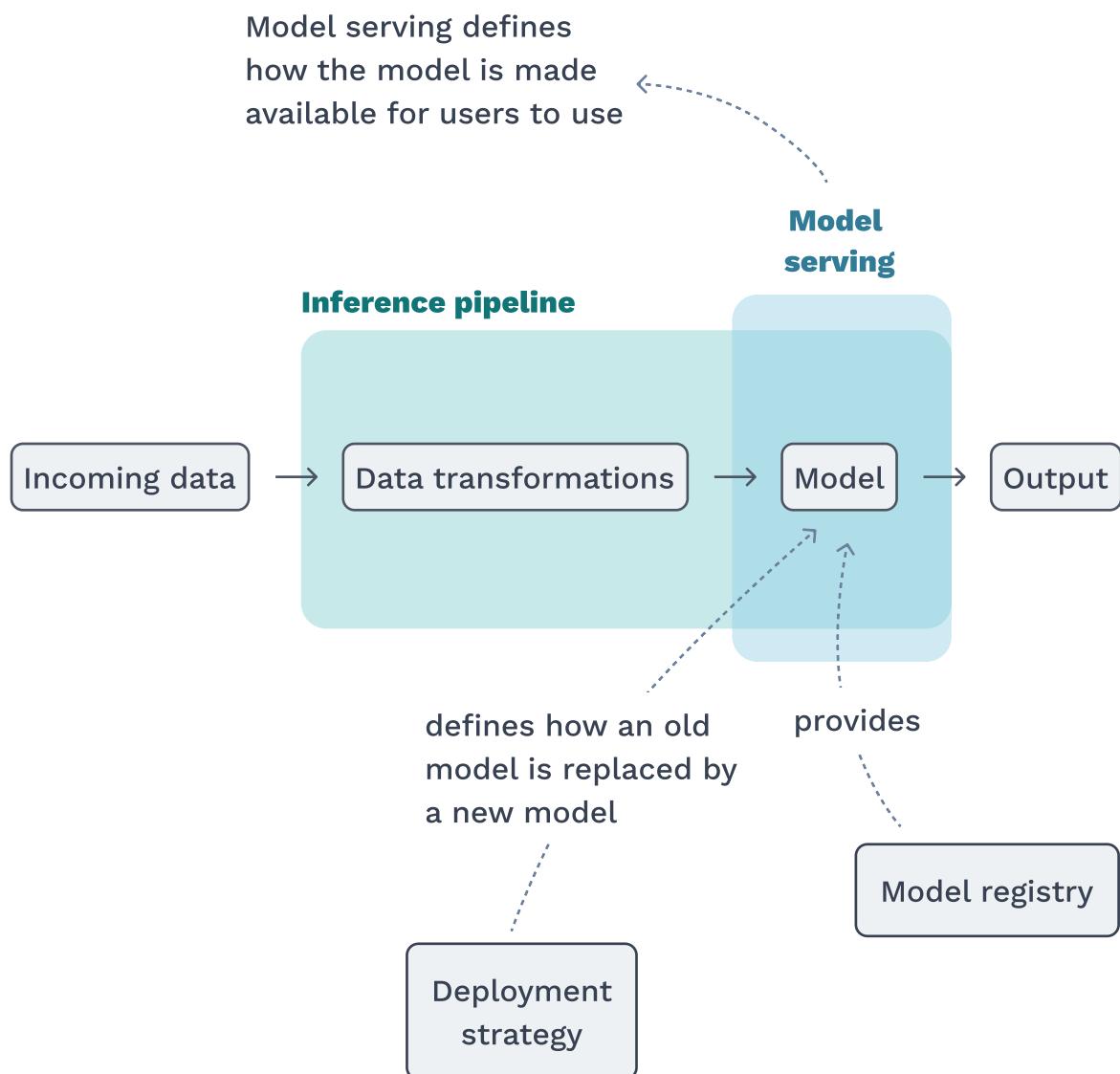
Overview of the deployment process

The Deployment process consists of several steps aimed at seamlessly promoting the model into the production environment. Model serving patterns are used so that the model can be accessed through the right medium depending on the particular use case. Deployment strategies are implemented to mitigate the potential risks associated with the introduction of a new model, while monitoring tools provide developers with insights into the performance of the models in the production environment.



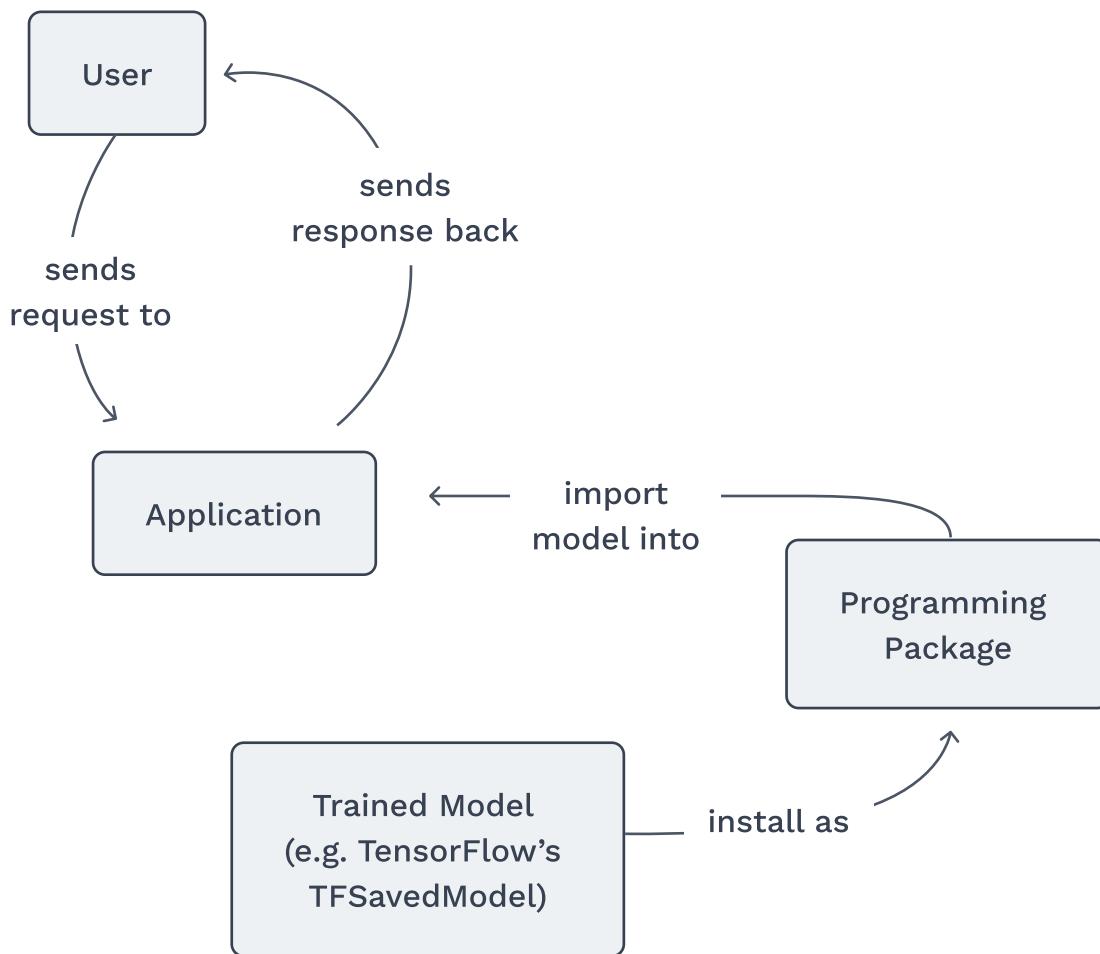
Inference pipeline

Once your model is deployed, it needs input data to make predictions. The results are then sent to users. Your data travels through an inference pipeline, following a path until it reaches the model, and the output it creates finishes the journey. The inference pipeline is a piece of software that takes production data as input, transforms it and makes predictions with the current model in production.



Model serving patterns • Model as a dependency

The way in which a model is made available to users is referred to as the model serving pattern. Three patterns can be distinguished. On this page, we focus on the model as a dependency pattern. The core idea of this pattern is that a model is wrapped in a software application and served through this application. Try to identify at least two scenarios where this pattern should be used.

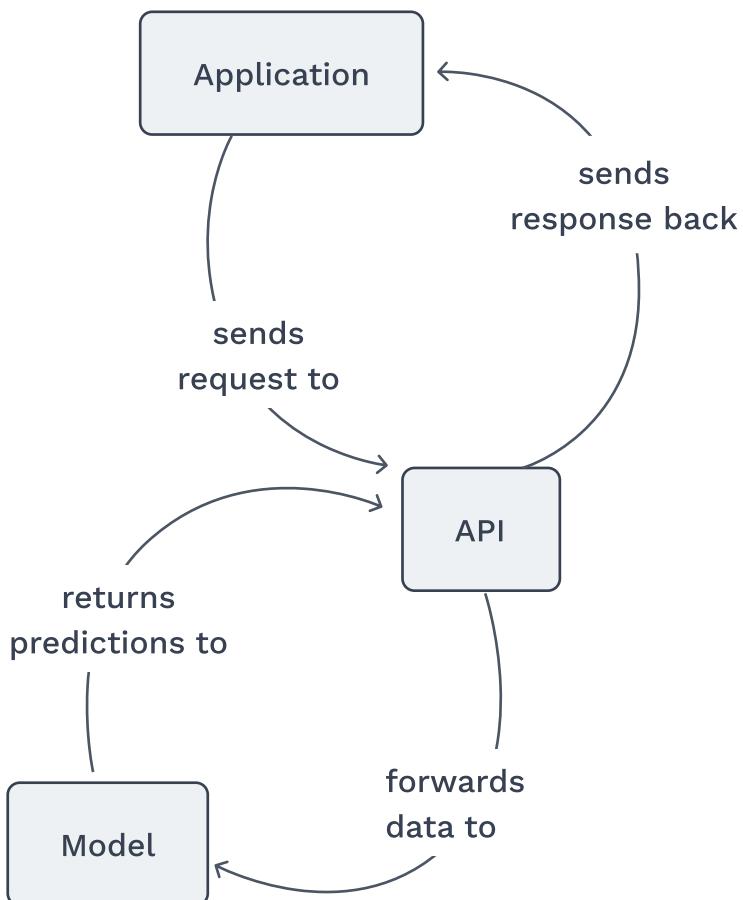


Resources

<https://ml-ops.org/content/three-levels-of-ml-software#code-deployment-pipelines>

Model serving patterns • Model as a service

The core idea of this pattern is that a model is served through an API and can be accessed by applications through a call to that API. Try to identify at least two scenarios where this pattern should be used.

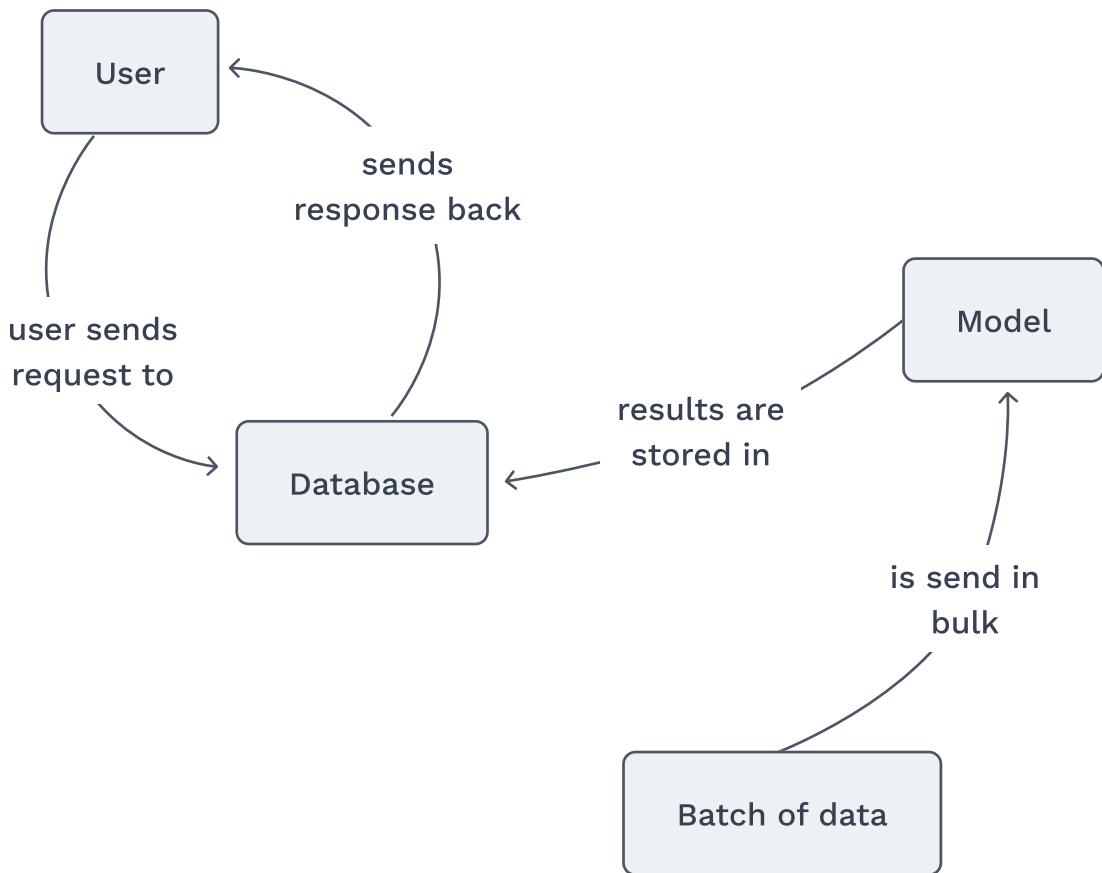


Resources

<https://ml-ops.org/content/three-levels-of-ml-software#code-deployment-pipelines>

Model serving patterns • Precompute service pattern

The core idea of this pattern is that all possible results of a model are pre-computed and stored in a database before a user makes a request. Try to identify at least two scenarios where this pattern should be used.



Resources

<https://ml-ops.org/content/three-levels-of-ml-software#code-deployment-pipelines>

Benefits of different model serving patterns

Picture yourself in charge of figuring out the most suitable model serving pattern. For the following three use cases, try to identify the ideal model serving pattern.

An online marketplace that sells cars wants to predict which cars a customer might want to buy. They plan to use a brief survey for this. The company knows all the possible options and wants to show the results as quickly as possible on their website.

- Precompute service pattern
- Model as a dependency
- Model as a service

An automobile company developed a model that it wants to use on its Engine Control Unit which is the “car’s computer”. The car is not always connected to the internet, the sensor data is continuous, and the hardware does not have a lot of data storage or compute.

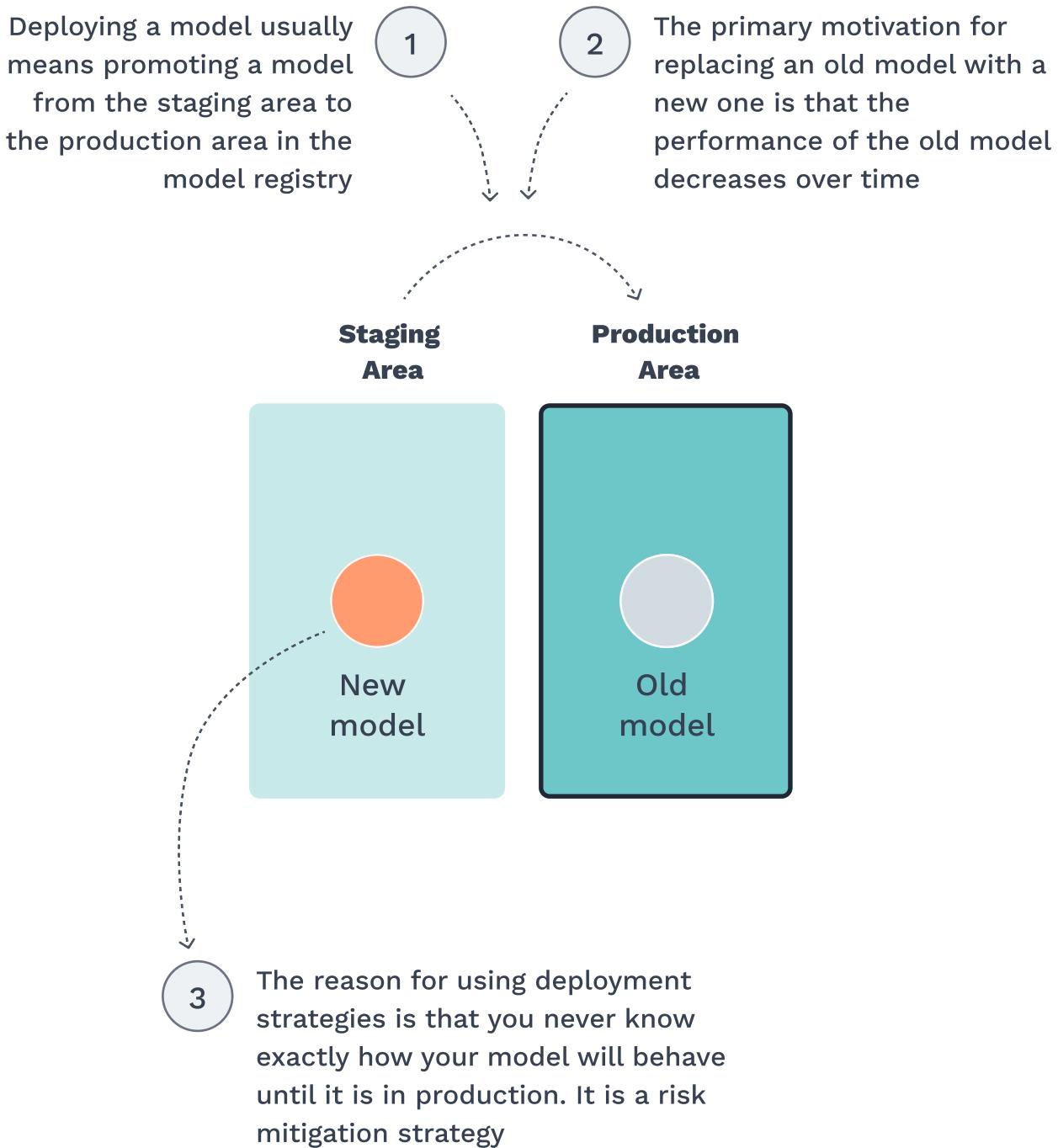
- Precompute service pattern
- Model as a dependency
- Model as a service

A music recommendation system tries to predict which song a listener would like based on streaming history. The data includes continuous data such as song length. The service is connected to the Internet and the model would need to be constantly updated to keep up with new songs and the changing tastes of the listener.

- Precompute service pattern
- Model as a dependency
- Model as a service

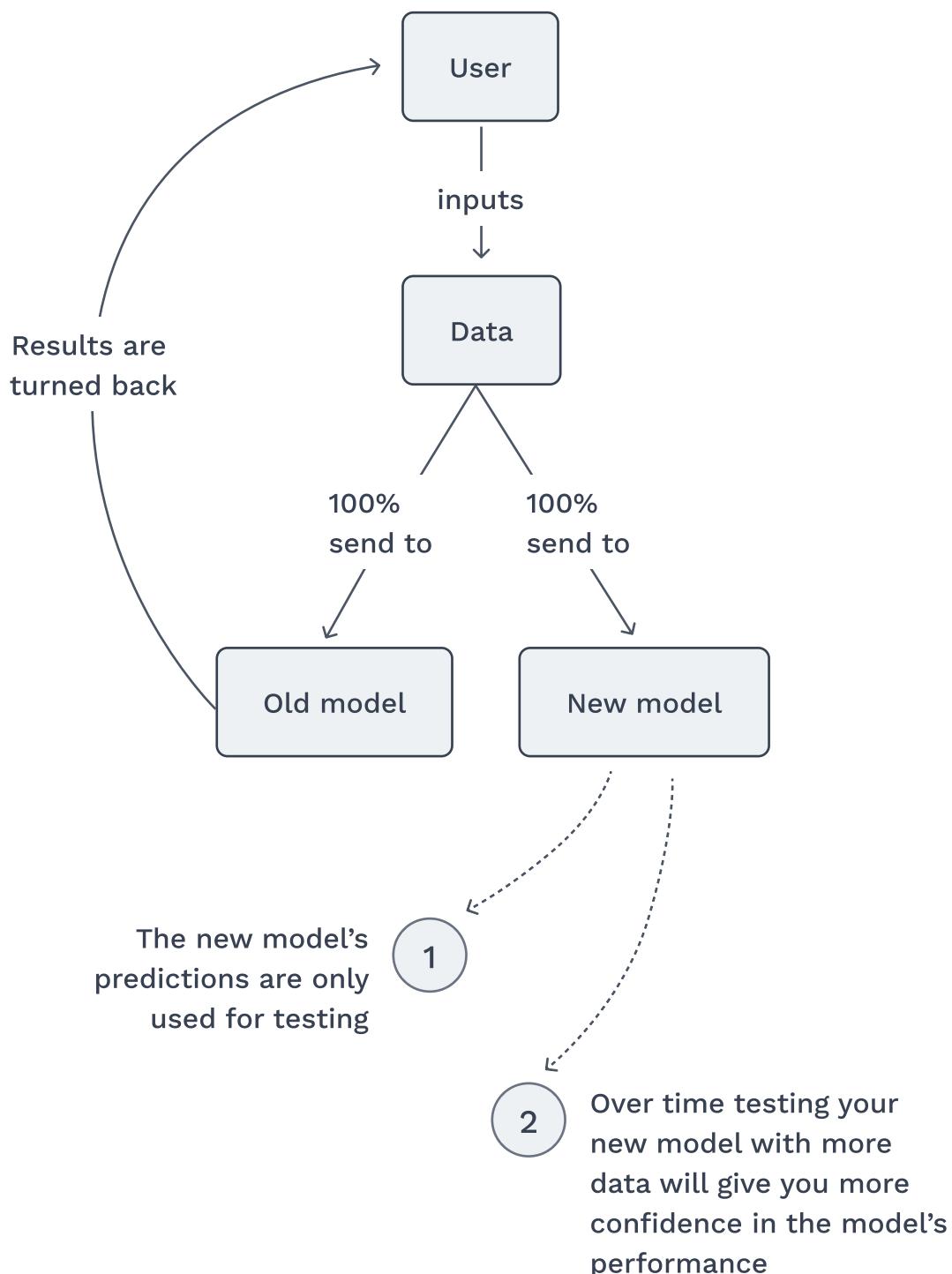
Why deployment strategies

At some point in the ML Lifecycle, you'll have to update an existing model with a new model. The way you swap out the old model for the new one is known as the deployment strategy.



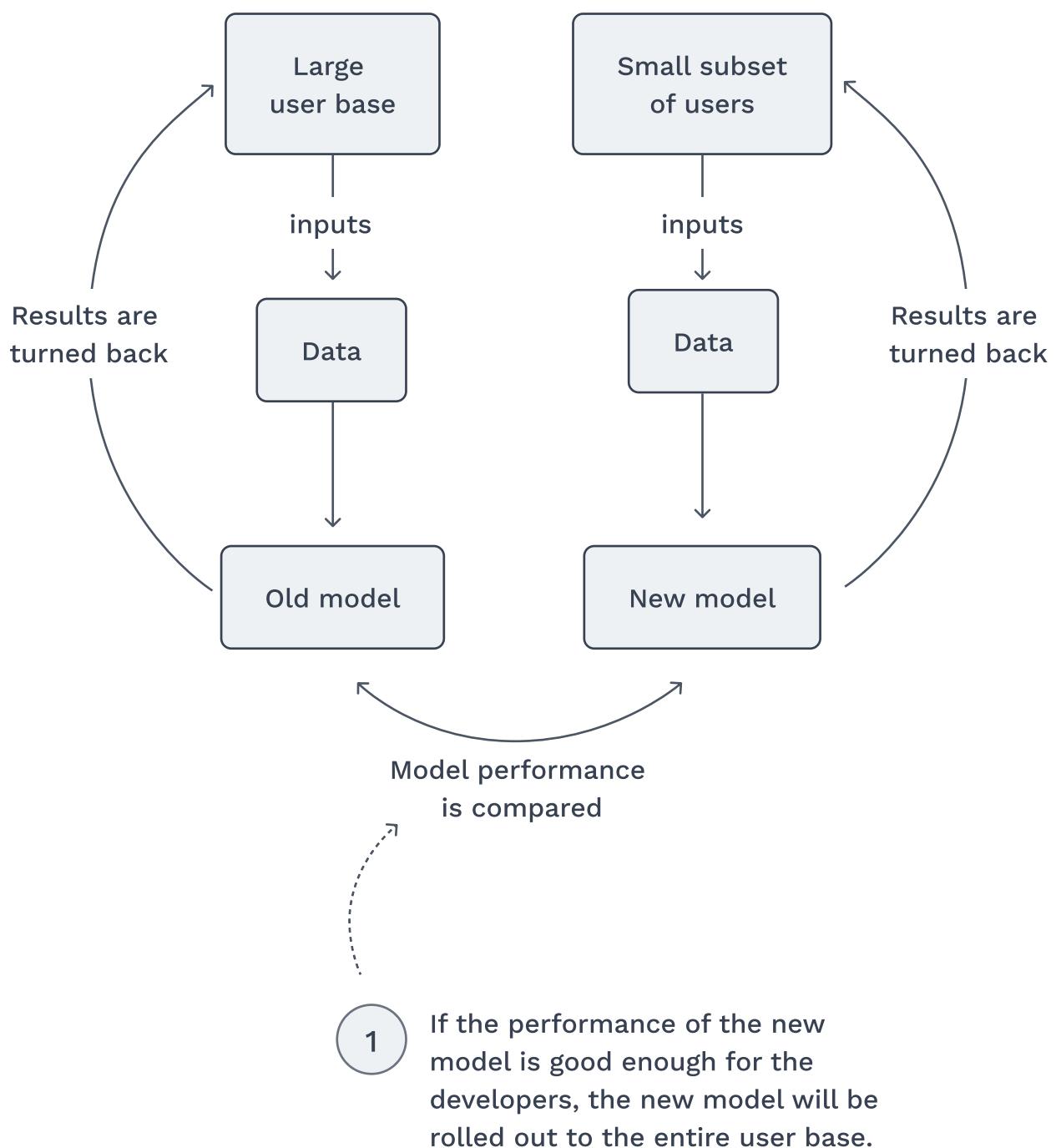
Shadow deployment

Shadow deployment is a strategy for high-risk use cases, such as clinical devices. The idea is to run input data over both the old and new models. However, the response from the new model is not served to the users and is only used for testing purposes. Only when our team is convinced that the performance of the new model is good enough, the new model will be deployed.



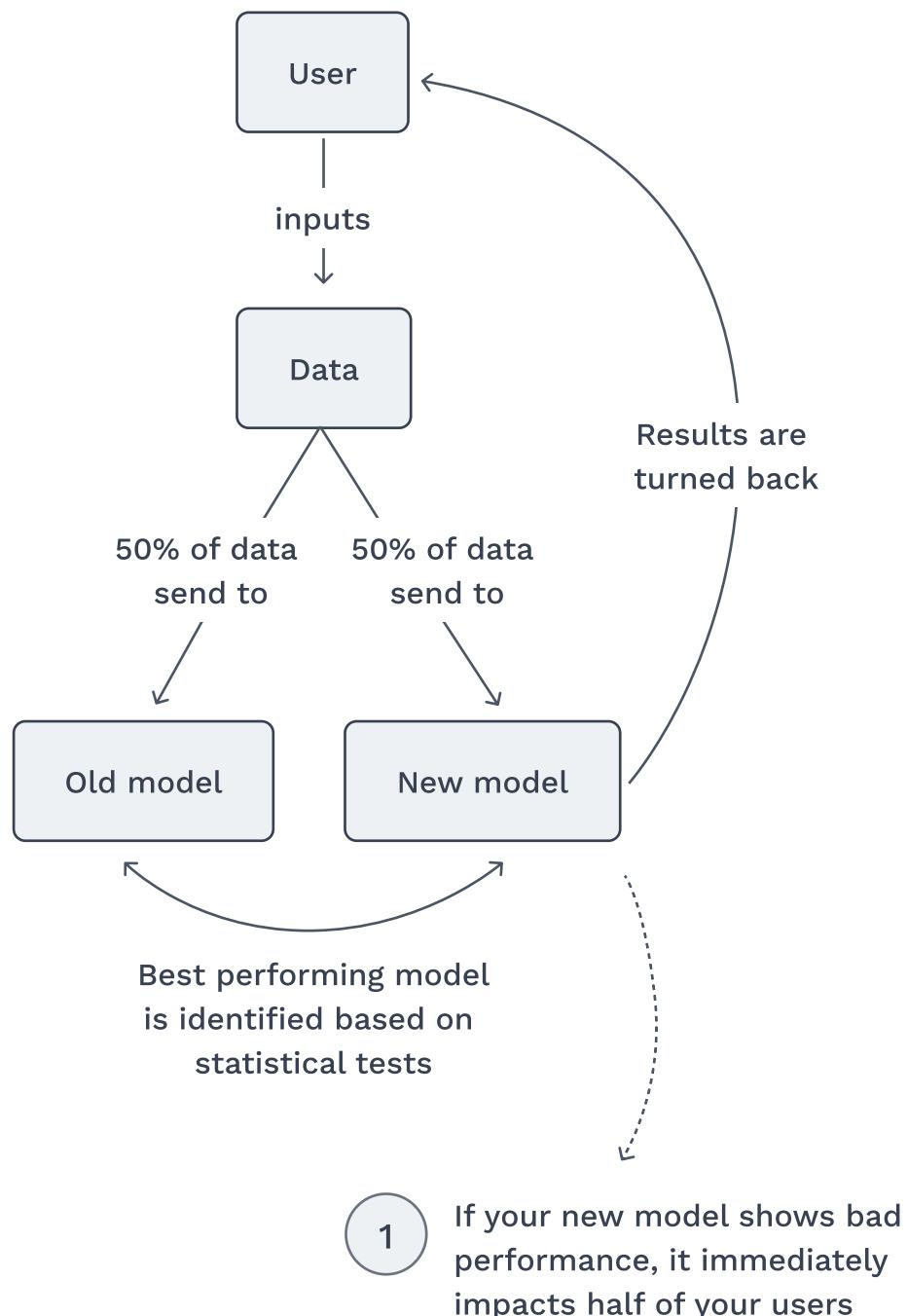
Canary deployment

With this strategy, your new models are rolled out to a small subset of users before deploying it to the entire user base. This strategy allows the developers to monitor the release for potential issues.



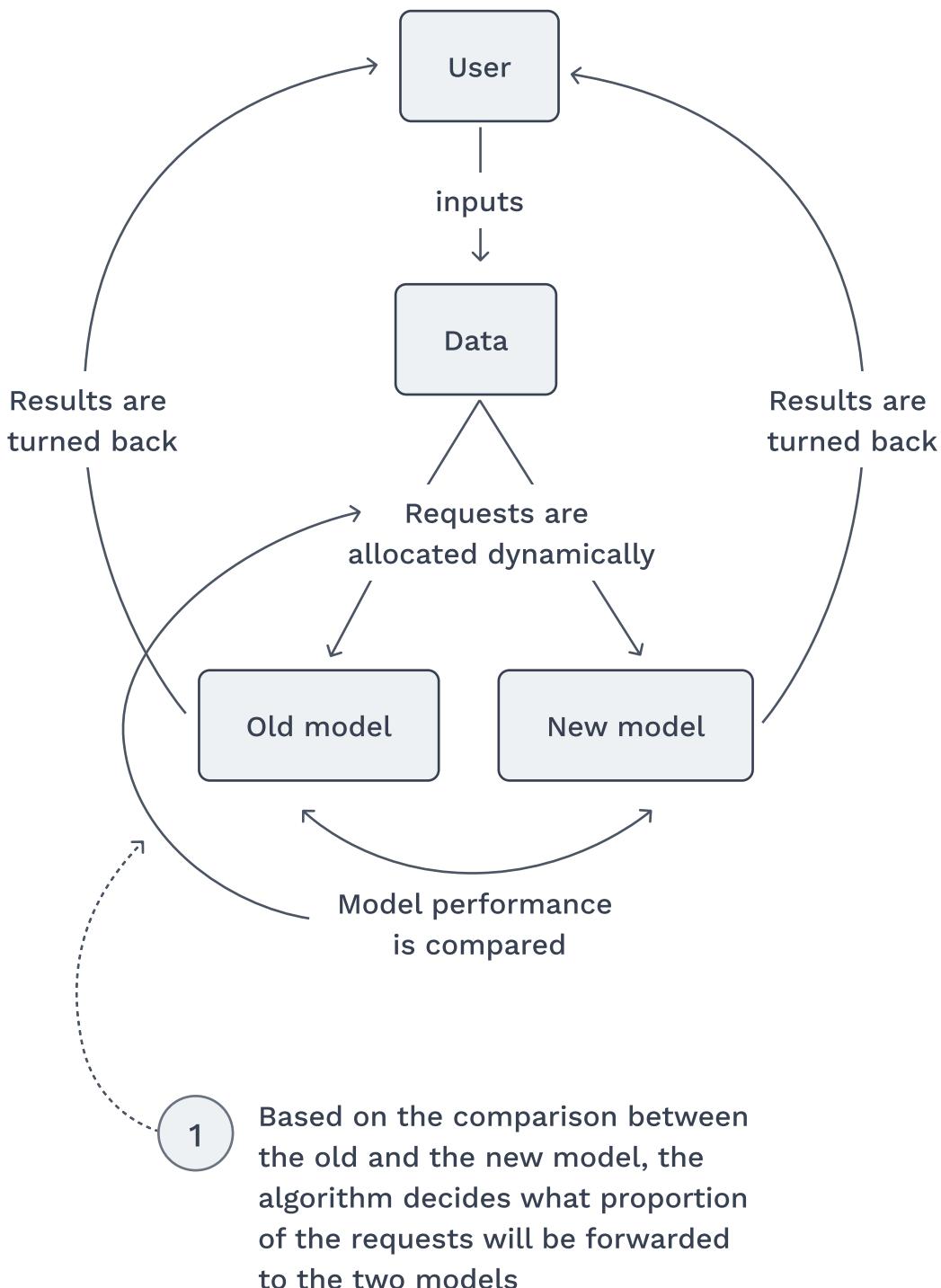
A/B testing

With this strategy, 50% of your users will be working with the new model right after you deploy it. It is therefore not suitable for high-risk use cases, but it has the advantage of giving you a quick insight into the production quality of your new model and how the users might interact with it.



Multi-armed bandit

This strategy is the only dynamic strategy among the three. Dynamic because it constantly changes the proportion of requests that go to the old model or the new model. Over time, and if the performance of the new model is better than the old, the majority of requests will go to the new model.



Which deployment strategy to use

Determining the appropriate deployment strategy often depends on a number of factors. Take a look at the following use cases and determine the most appropriate strategy for each.

A company plans to put out a new version of their model and quickly check if it works better than the old one. They want to make sure it doesn't cause any big problems. If it makes mistakes, it won't hurt anyone, but it might make a few users unhappy with the service.

- A/B testing
- Shadow deployment
- Multi-armed bandit

A financial institution wants to test a new fraud detection model without risking disruption to its current operations. The new model runs in parallel, and its predictions are compared with the existing system to assess its accuracy and reliability.

- A/B testing
- Shadow deployment
- Multi-armed bandit

A media streaming platform wants to use ML models that predict longer viewing times. They want to dynamically provide more traffic to models that lead to longer viewing times.

- A/B testing
- Shadow deployment
- Multi-armed bandit

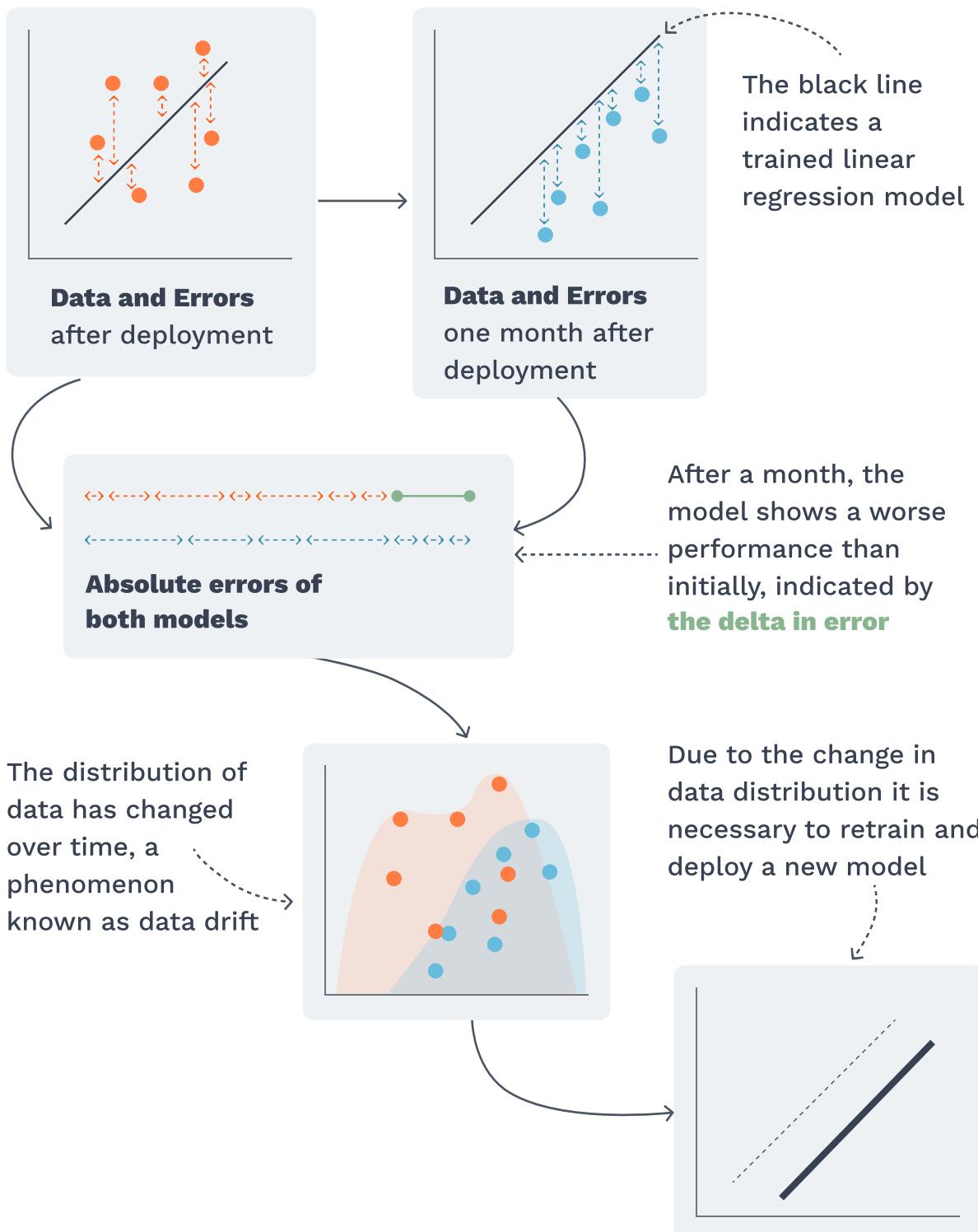
Without monitoring your ML system will fail

Once you deploy the model, your job goes beyond the initial deployment. Regular monitoring is important to keep the model up to date. Monitoring shouldn't be an afterthought in ML system development; instead, it's an essential step to ensure your model stays accurate and relevant over time.



Data drift • A common target for monitoring

Data drift refers to the phenomenon where the statistical properties of a dataset change over time, leading to a degradation in the performance and reliability of machine learning models. It occurs when the underlying data distribution changes, causing discrepancies between the training and operational data.



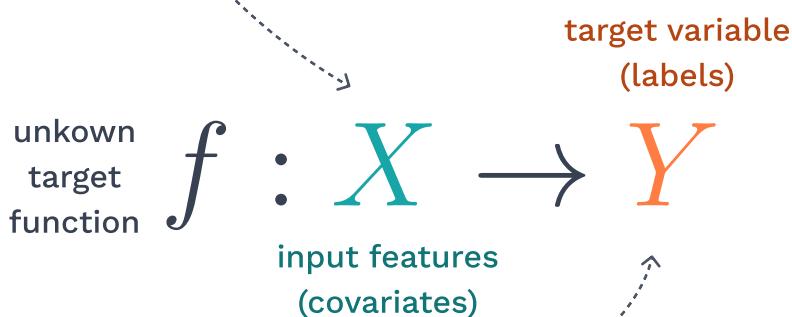
Covariate shift and label shift

Data drift can be divided into two types: covariate shift and label shift. Both can affect the performance of models by causing differences between the data used for training and the data encountered during deployment.

1

Covariate shift

Covariate shift refers to a situation where the distribution of the input features (covariates) in the training data differs from the distribution of the input features in the production data. In other words, the statistical properties of the input space change between the training and production phases.



2

Label shift

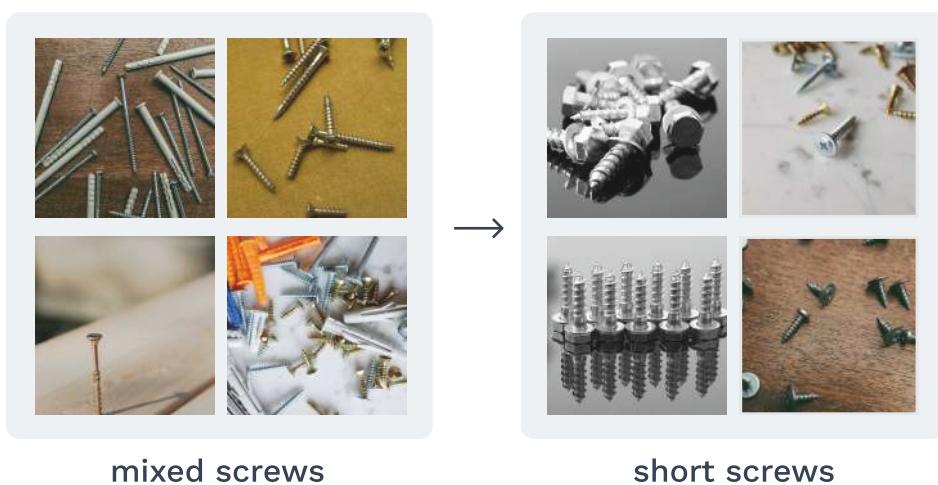
Label shift in machine learning refers to a situation where the distribution of the target variable (labels) changes between the training and deploying phases of a model. This change can lead to a discrepancy between the training and real-world data, making the model's predictions less accurate or biased.

Examples of covariate and label shift

Below you can find an example for a covariate and a label shift each.

Covariate shift

In this scenario, a model was trained to distinguish between screw types. Initially, the product range included screws of different lengths. The model used images of screws as a feature. Over time, however, the screws became shorter, which led to a deterioration in the accuracy of the model.

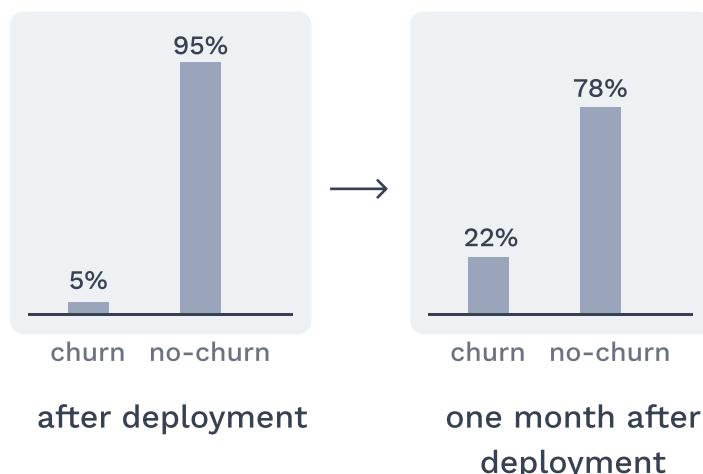


mixed screws

short screws

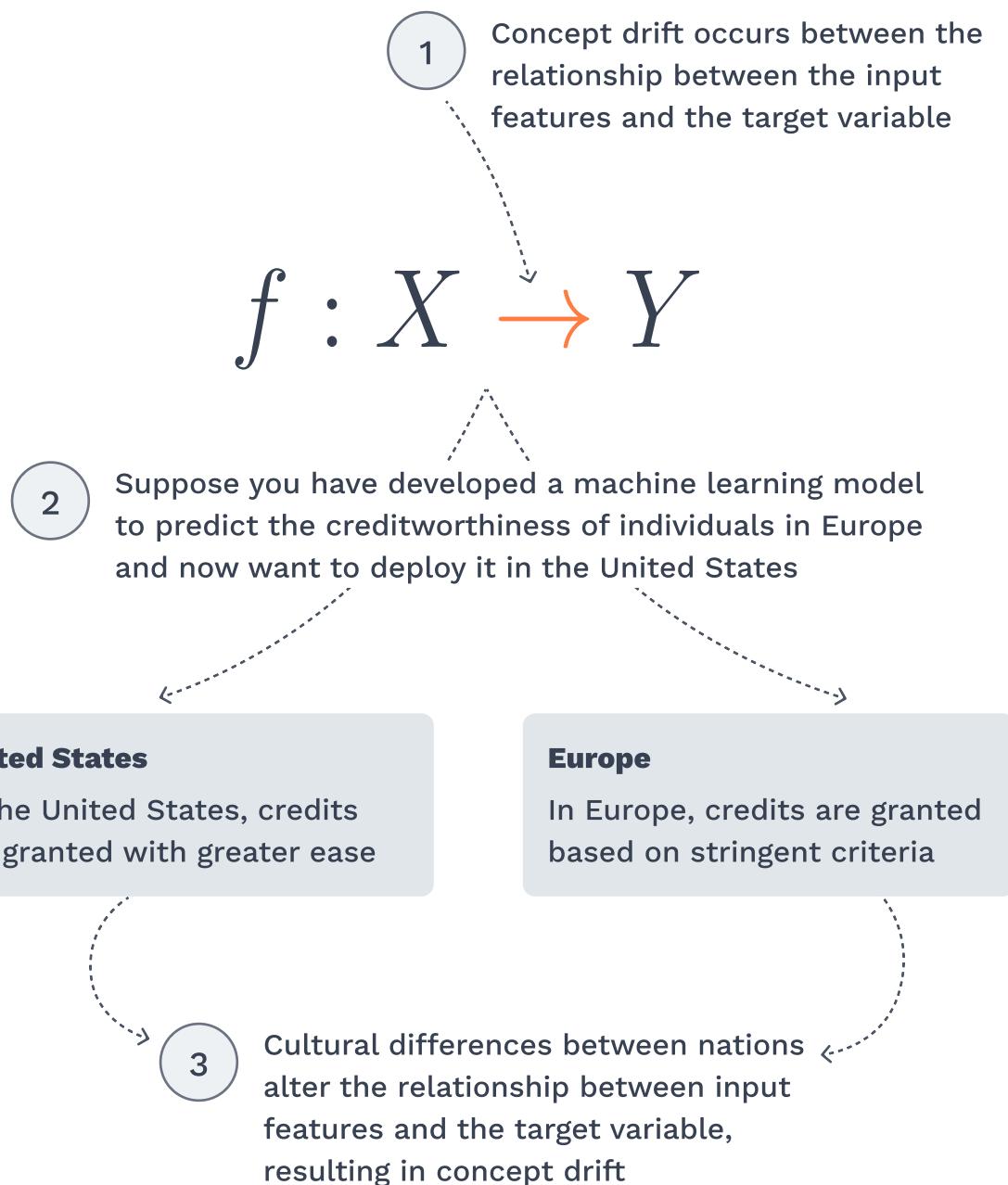
Label shift

In this example, a model underwent training to forecast customer churn. Over time, there has been a noticeable shift in the distribution of the target variable. Specifically, there has been a 17% increase in the number of customers who churned compared to the previous month when the model was deployed.



Concept drift • Shifts in relationship patterns

Concept drift refers to the phenomenon where the underlying relationship between input features (covariates) and the target variable (labels) in a machine learning problem changes over time, leading to a degradation in model performance.



Resources

<http://www.trustworthymachinelearning.com/trustworthymachinelearning-09.htm>
<https://towardsdatascience.com/mlops-model-monitoring-101-46de6a578e03>

Concept drift, covariate shift, or label shift?

Try to identify which of the three drifts and shifts occurs in each of the following cases. Pick one of the three concepts in each case.

A computer vision model trained on images captured by high-quality cameras fails to generalise well when deployed in real-world scenarios where images are captured by low-quality or different types of cameras.

- Concept drift
- Covariate shift
- Label shift

An online marketplace used a model to predict user buying behaviour for product categories. Originally, sales were split 50% electronics and 50% cooking appliances. Recently, however, customers purchased an average of 90% of products from the cooking appliances category.

- Concept drift
- Covariate shift
- Label shift

In a weather prediction model, the relationship between atmospheric pressure and precipitation patterns changes due to climate change, making the model's predictions less accurate over time.

- Concept drift
- Covariate shift
- Label shift

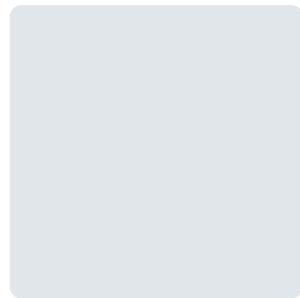
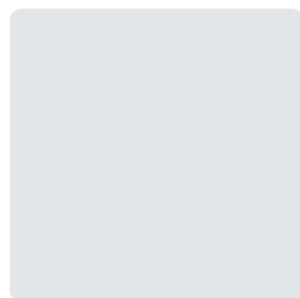
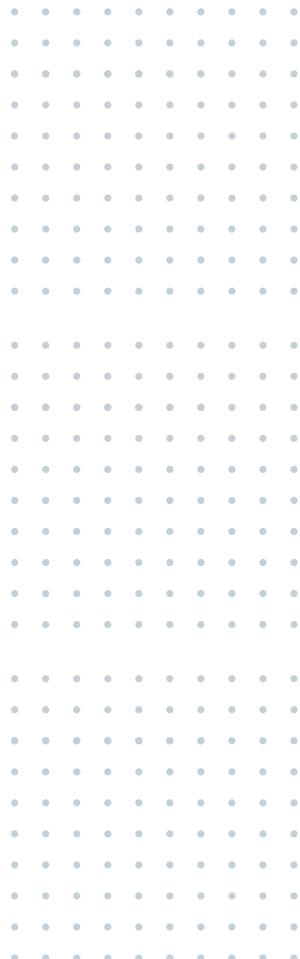
Types of concept drift

Concept drift can occur quickly, gradually, or in specific patterns. Below you can find six boxes. We have given you an example. Try drawing other possible patterns and give an example for each.

Instant



A sudden change that has long lasting effects like Covid.



Resources

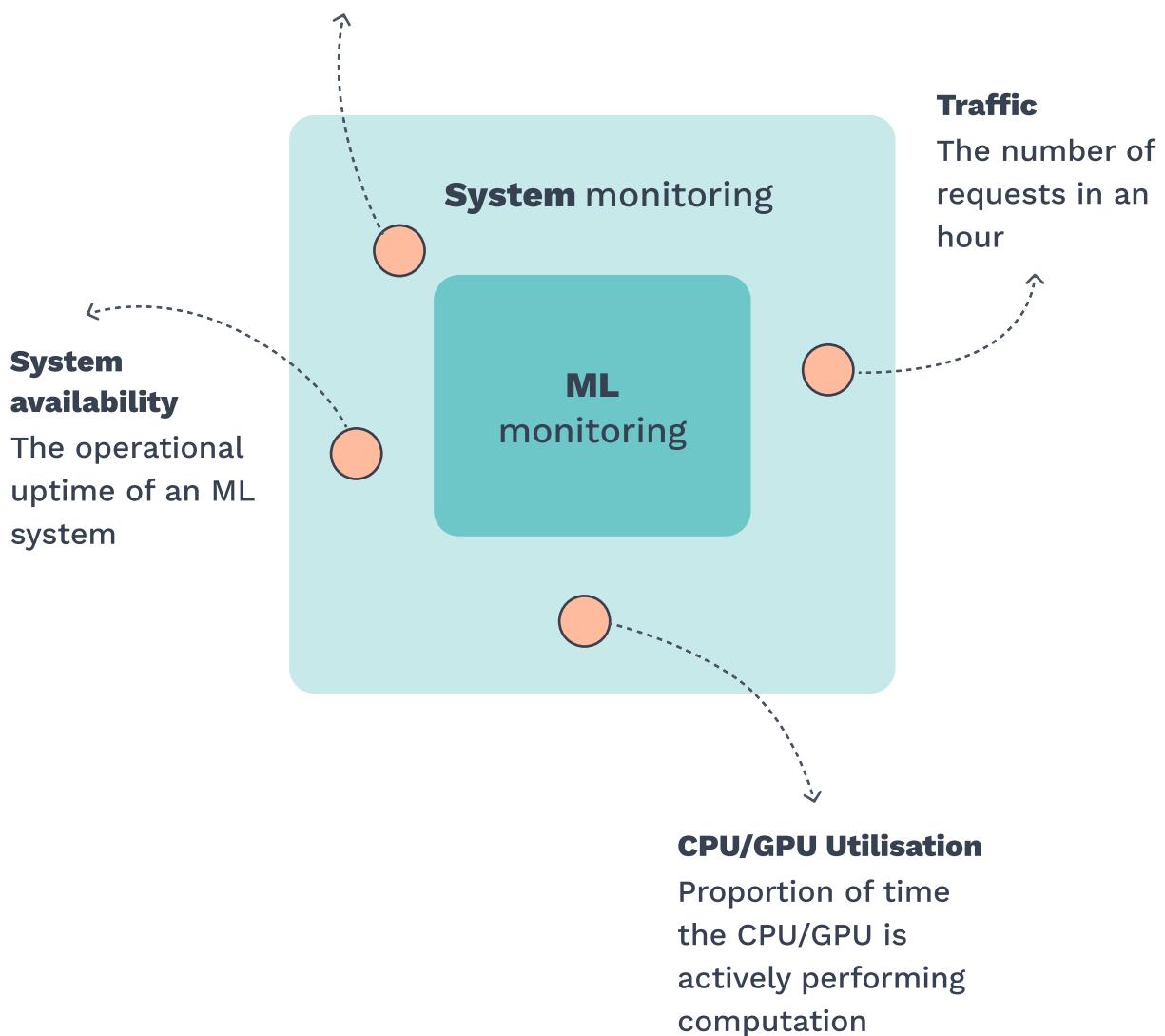
<https://www.iguazio.com/blog/concept-drift-deep-dive-how-to-build-a-drift-aware-ml-system/>

System monitoring • Changes on the system level

System monitoring refers to the process of tracking changes in system metrics in a production environment. The term "system" means that these metrics are not specific to the ML model itself, but relate to the overall functionality of the machine learning system.

Latency

Time delay or the period it takes for a system to respond to a request



Practical recommendations

If you're uncertain about what to do next, consider these practical recommendations.

The deployment of your models is not the end of your ML Lifecycle. When planning your project, try to budget monitoring. Proper monitoring can be costly, but it is critical to deliver models that produce quality predictions in the long run.

When deploying your models, you should consider both model serving patterns and deployment strategies. There is no universal solution, but the specific use of both depends on your use case.

Things can go wrong. Always make sure that you can quickly rollback to a previous model version. This is especially important for high-risk use cases.

As you become more familiar with model serving patterns and deployment strategies, consider using more complex systems such as cascading model patterns or ensemble patterns.

Self-assessment of comprehension

Take a moment to answer the following questions about the content covered in this module. Keep in mind that there's only one correct answer for each question.

1. Which of the following list of shifts and drifts has the correct terminology?

- A) data drift, concept shift, covariate shift, label shift
- B) data shift, concept drift, covariate drift, label shift
- C) data drift, concept drift, covariate shift, label shift

2. Which of the following analogies best describes A/B testing?

- A) Imagine you're a filmmaker trying to gauge audience preferences for two different endings of your movie. You show one ending to a group of viewers at a cinema and the other ending to another group watching it on a streaming platform. By analyzing their reactions, ticket sales, and streaming views, you can determine which ending resonates better with the audience.
- B) Imagine you're a radio station manager trying to determine which song to play during peak hours. You have two options: Song A and Song B. To gather data on listener preferences, you randomly select periods throughout the day to play one of the two songs. Based on the number of requests and engagement from listeners during each period, you make a decision on which song to play more frequently.
- C) Imagine you're a fashion designer launching a new collection. Before releasing it to the public, you organize a private fashion show for a select group of individuals. They provide feedback on the designs, suggesting improvements or pointing out any issues they notice. This allows you to refine the collection and address any concerns before making it available to a wider audience.

Self-assessment of comprehension

3. What is a commonality between A/B testing and multi-armed bandit?

- A) Both methods rely on machine learning techniques to analyse model performance.
- B) Both methods aim to minimise model bias.
- C) Both methods involve user experimentation to determine the best models.

4. What is the key idea to distinguish between label shifts and covariate shifts? Remember that both shifts describe changes from training data to production data.

- A) Label shift refers to changes in the features or attributes of the data, while covariate shift refers to changes in the target variable.
- B) Label shift occurs when there are changes in the distribution of the input features, while covariate shift occurs when there are changes in the distribution of the target variable.
- C) Label shift refers to changes in the distribution of the target variable, while covariate shift refers to changes in the distribution of the input features.

Self-assessment of comprehension

5. Which of the following scenarios best describes a “model as a service” architecture?

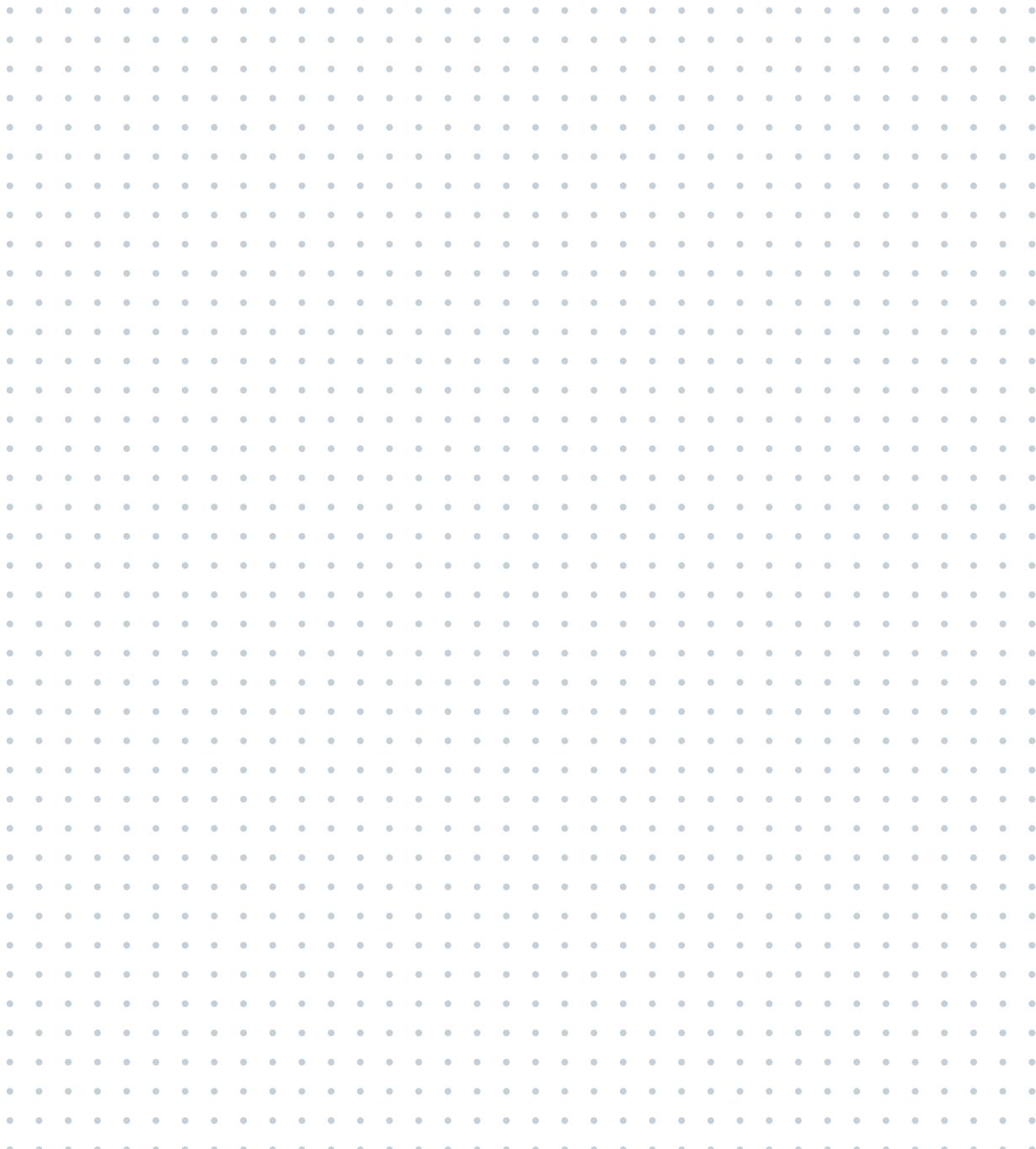
- A) A financial institution deploys a fraud detection system that analyzes customer transactions in real time to identify potential fraudulent activity. The model is trained using historical transaction data and continuously updated with new data to adapt to evolving fraud patterns. Whenever a customer initiates a transaction, the model is invoked and provides a fraud probability score. The model is called from within the application and loaded as a software package.
- B) An automobile is developed a model that it wants to use on its Engine Control Unit which is the “car’s computer”. The car is not always connected to the internet and does not have a lot of data storage or compute.
- C) A movie recommendation system attempts to predict which film a viewer would enjoy based on their viewing history and recommends it for their next watch. The model would require continuous updates to accommodate the release of new movies and the viewer's evolving preferences.

6. What is the difference between model serving patterns and deployment strategies?

- A) Model serving patterns primarily focus on selecting and orchestrating various components required to operationalise models in a production environment, while deployment strategies encompass the design patterns and methodologies for efficiently exposing machine learning models as APIs.
- B) Model serving patterns refer to the architectural patterns used to serve ML models, while deployment strategies involve the techniques to test and compare model performance in order to promote a model to production.
- C) Model serving patterns involve the selection and orchestration of various components required to operationalise models in a production environment, whereas deployment strategies primarily deal with data preprocessing techniques and feature engineering for machine learning models.

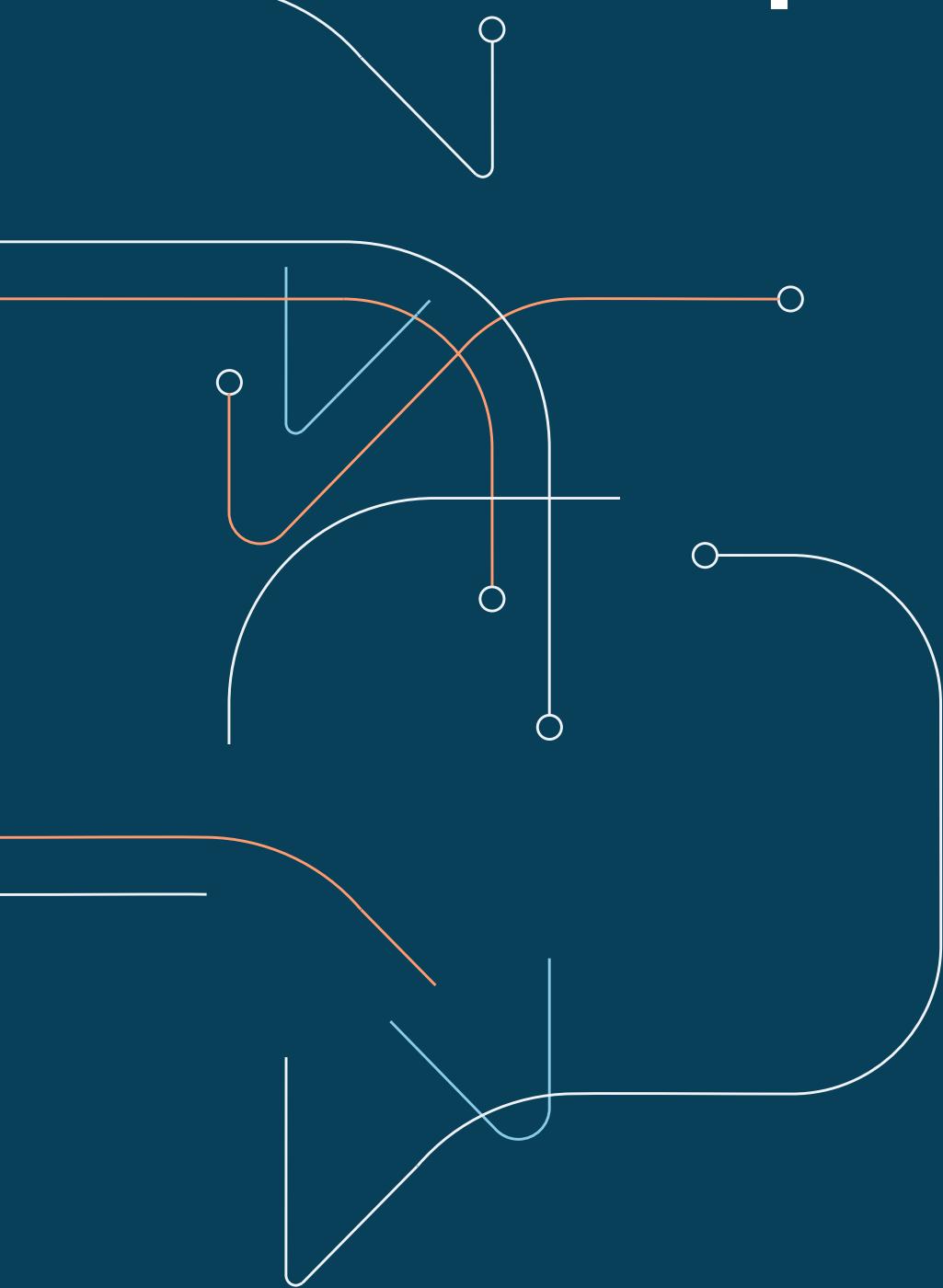
Module review

Each module begins and ends with this page. At the beginning of the module, we asked you to write down some questions that you would like answered at the end. Once you have written down these questions and worked through the content, take some time to look at the questions again and explain the answers to yourself.

A large grid of grey dots on a white background, intended for handwritten notes. The grid consists of approximately 20 columns and 30 rows of dots, providing a structured area for writing.

0 6

ML Orchestration & Feedback Loops



Intended learning objectives

This module is designed to improve your ability to effectively use a workflow orchestration tool and manage feedback loops within an ML system. First, we will define the terminology for workflows and workflow orchestration tools and then discuss the four main functions of these tools in detail.

By the end of this module, you will have developed the following proficiencies:

- ✓ Distinguish between the terms workflow and workflow orchestration tools.
- ✓ Describe the functionality and purpose of a workflow orchestration tool.
- ✓ Describe how workflow orchestration tools improve collaboration among team members.
- ✓ Describe how CI/CD can further professionalize your ML workflow.
- ✓ Describe four triggers for retraining models.
- ✓ Identify and name at least two long-range feedback and wide-range feedback loops.

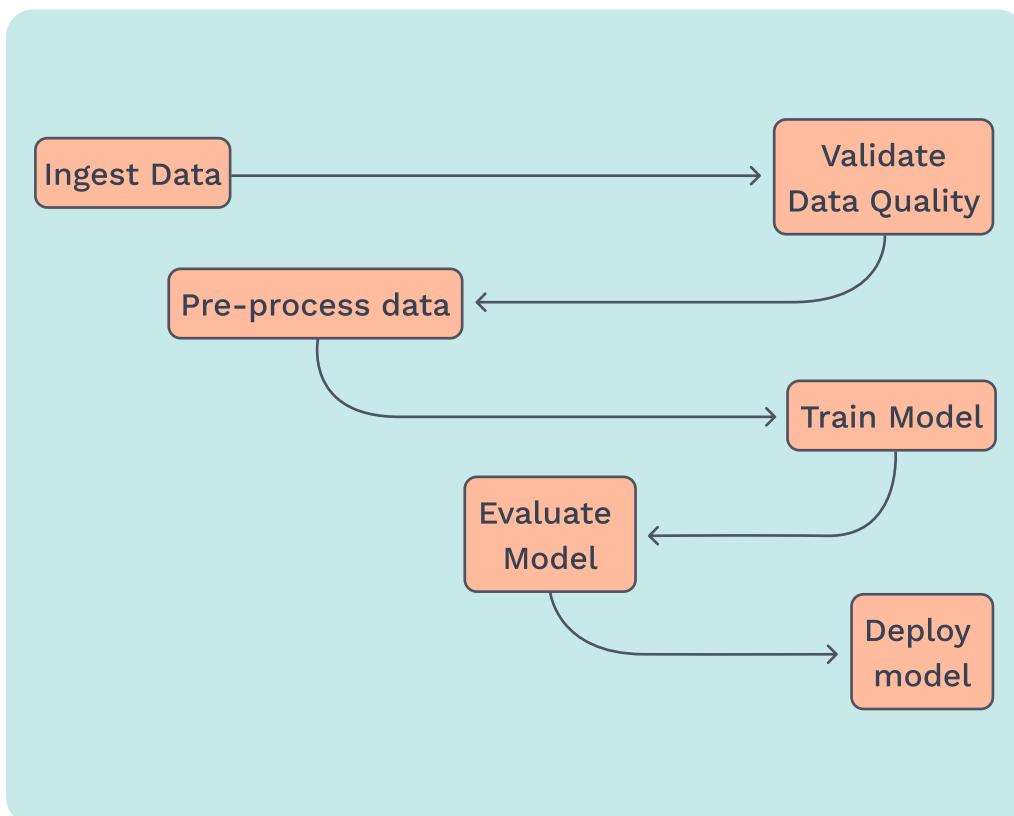
Before you begin, we recommend that you take a look at the contents of the module. Take a few minutes to go through each page and look at the headings and visuals. Try to get an overview of the module's content. When you are done, go to the last page of the module and write down a few questions you would like to have answered at the end of the module. We will ask you to review these questions later after you have worked through the content.

Definition of a workflow

An ML workflow describes how the ML Lifecycle is implemented by an ML team. While there are numerous activities within the ML Lifecycle, only a subset of them are executed in a particular workflow.

1

A workflow describes the sequence in which the activities for a particular use case are executed automatically or manually

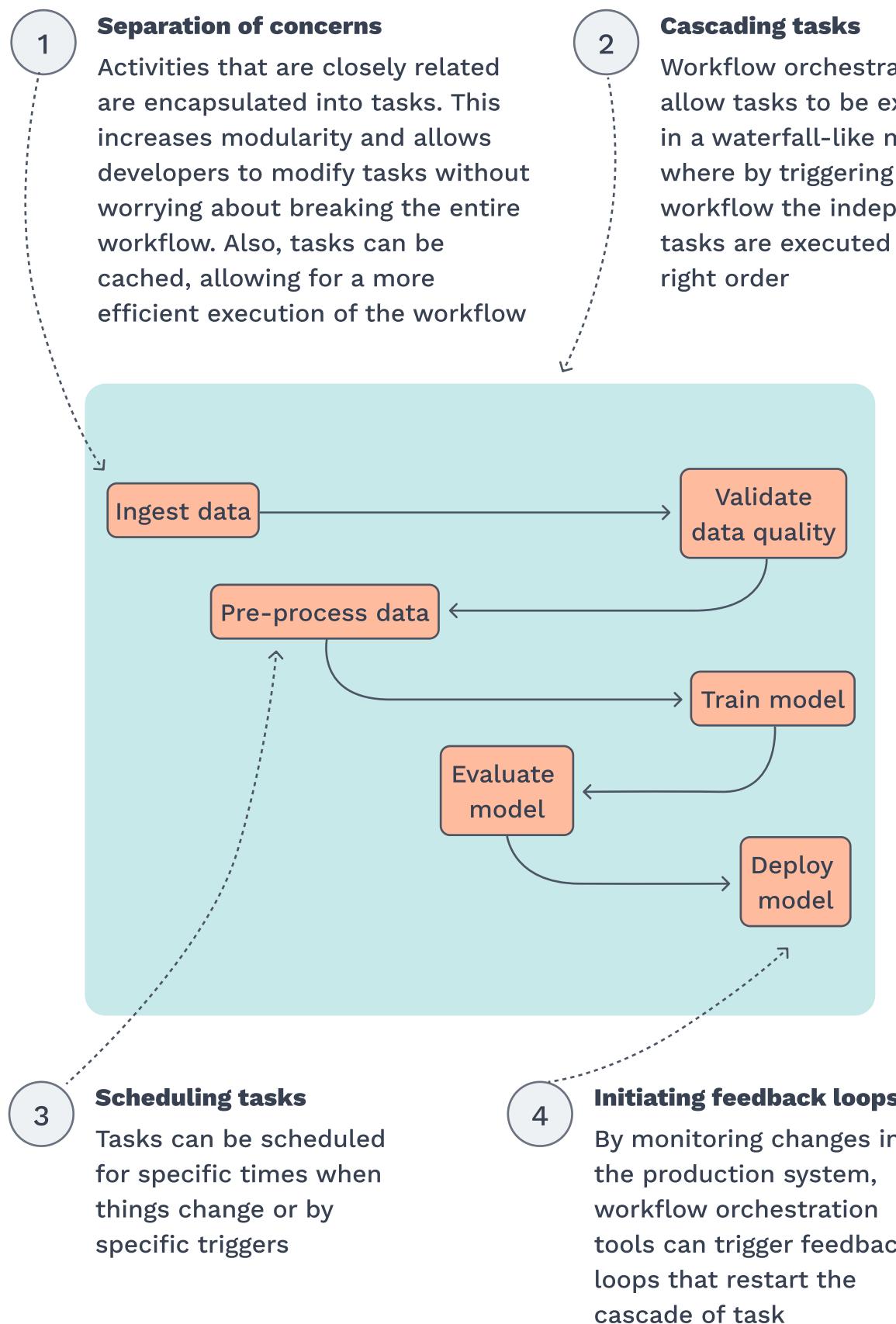


2

Your team has already implemented a workflow to train and deploy models. However, many workflows undermine the ML Principles

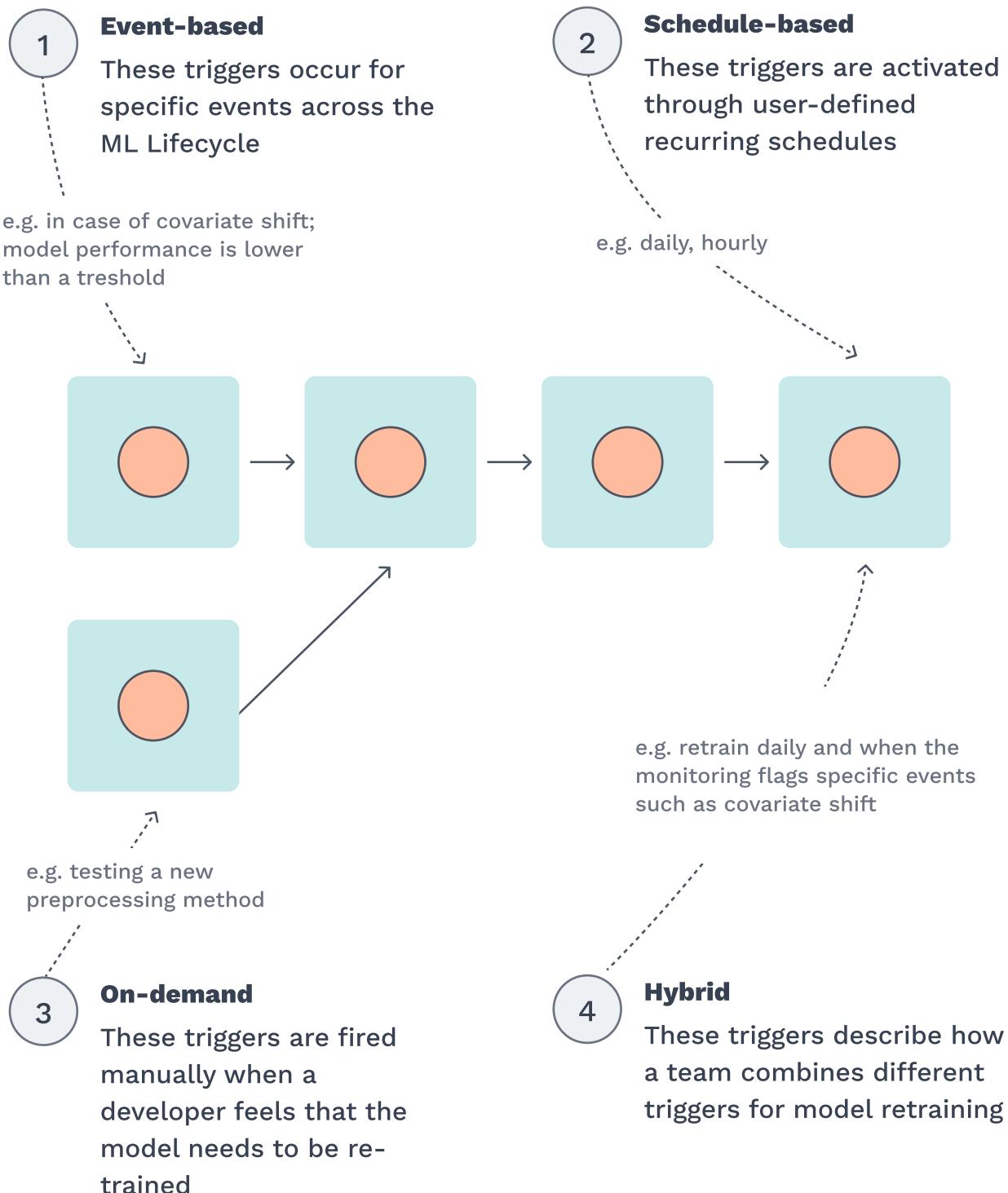
What does it mean to orchestrate a workflow?

Workflow orchestration tools are designed to automate and manage complex ML workflows. They perform four functions.



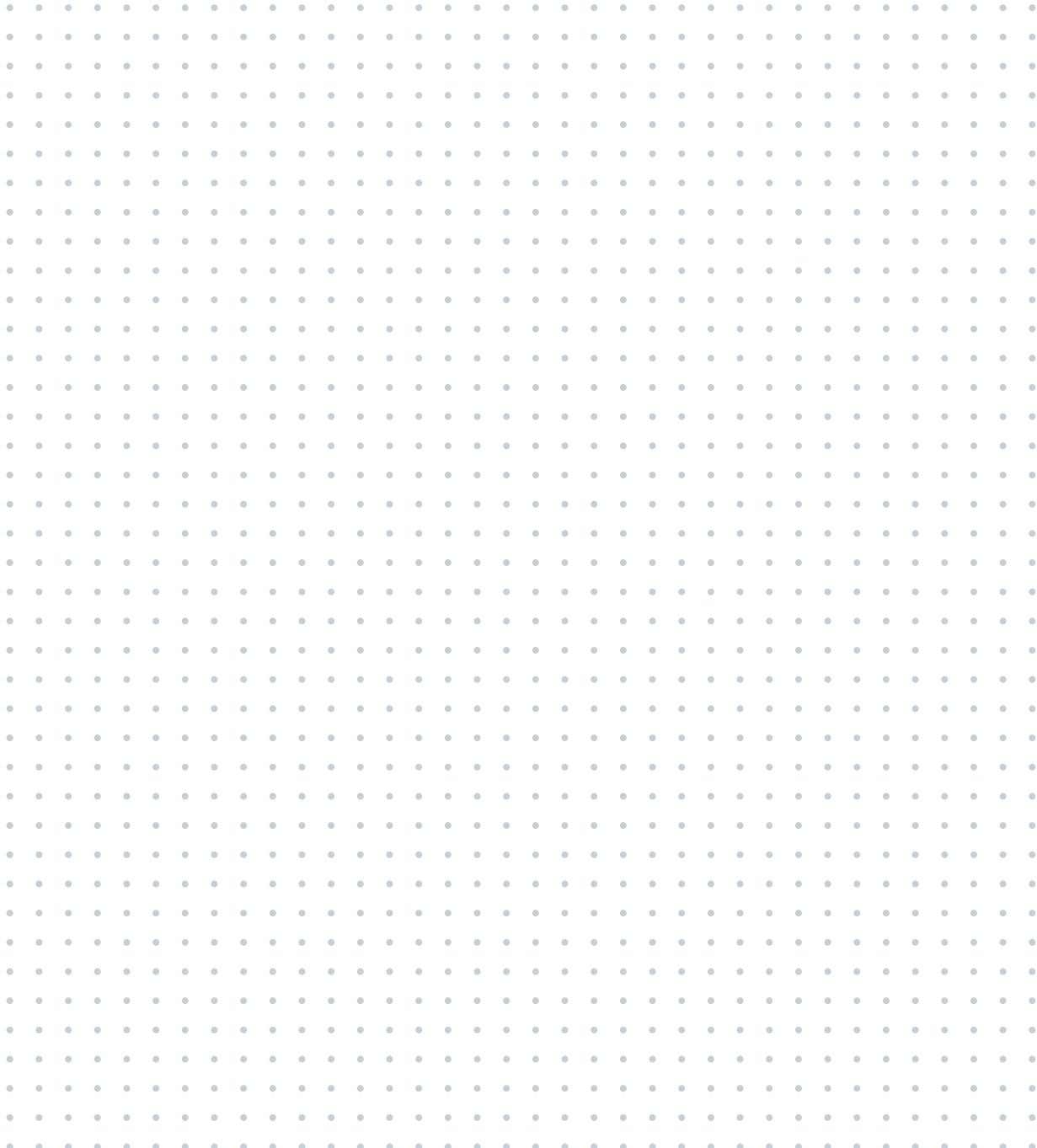
The four types of model retraining triggers

The feedback loops can be triggered in a variety of ways: action based, schedule based, on-demand and hybrid.



How workflow orchestration tools promote the ML Principles

Based on what we have said so far about workflow orchestration tools, try to describe which ML Principles are supported by orchestration tools and why? Select the principles that you think are particularly supported.

A large grid of gray dots on a white background, resembling a sheet of dot-grid paper. The grid is approximately 20 columns wide and 30 rows high, providing a space for writing or drawing.

The benefits of workflow orchestration tools from the perspective of the accountabilities

A workflow orchestration tool has one major advantage: it improves collaboration between team members. Try to predict which questions the different accountabilities can answer with a workflow orchestration tool.

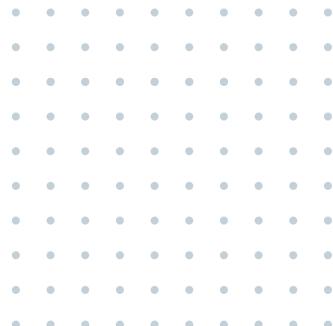


Product Owning
Paige

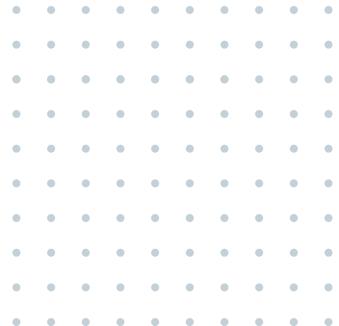
What is the
current status
of the
workflow?



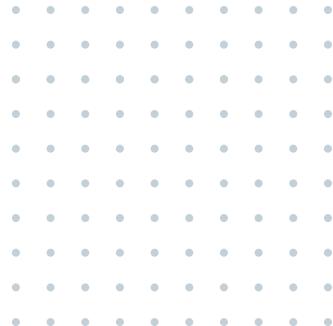
Data Science
Doris



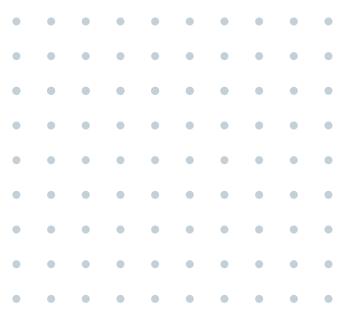
ML Engineering
Matthew



Data Engineering
Damian

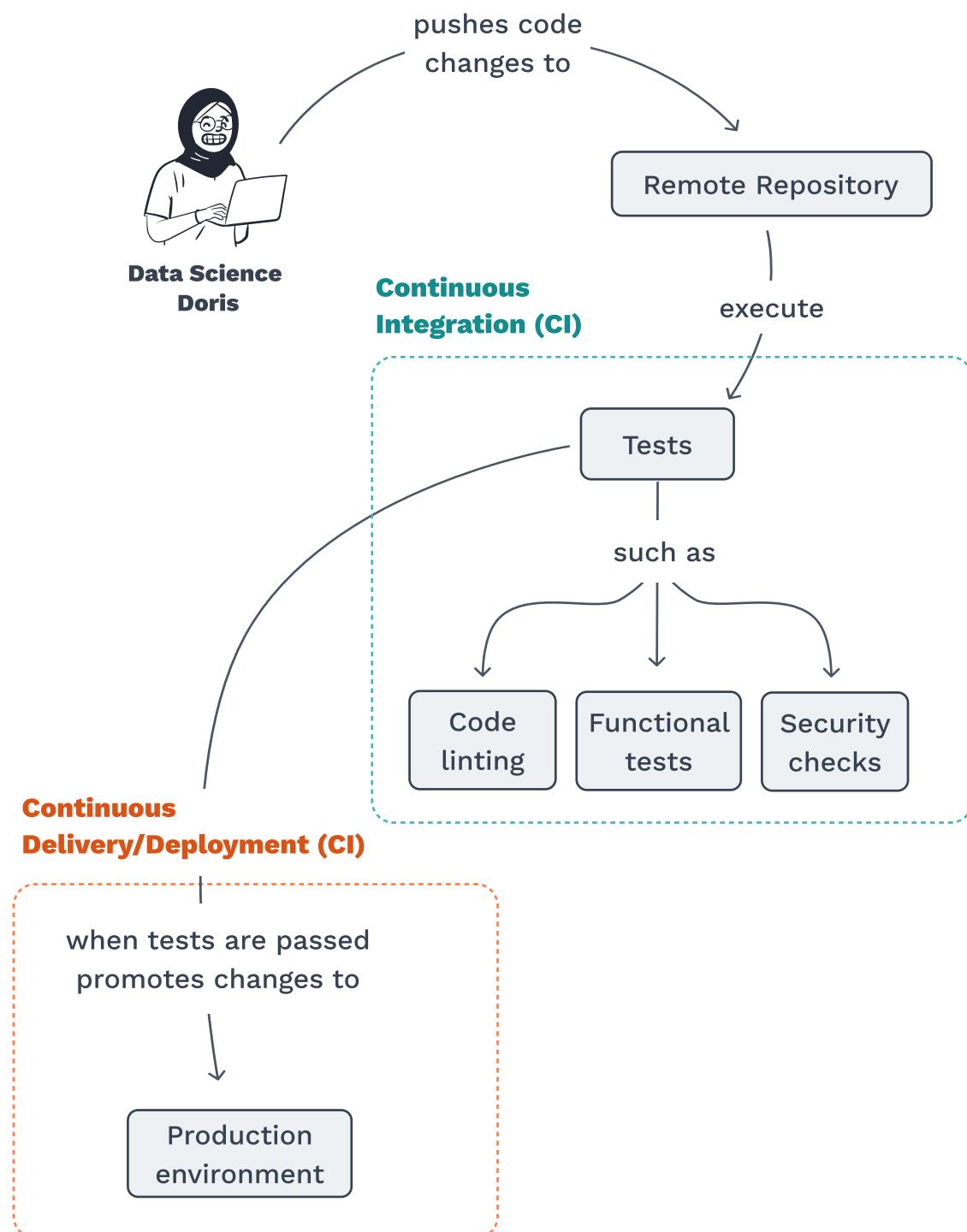


Data Steward
Deborah



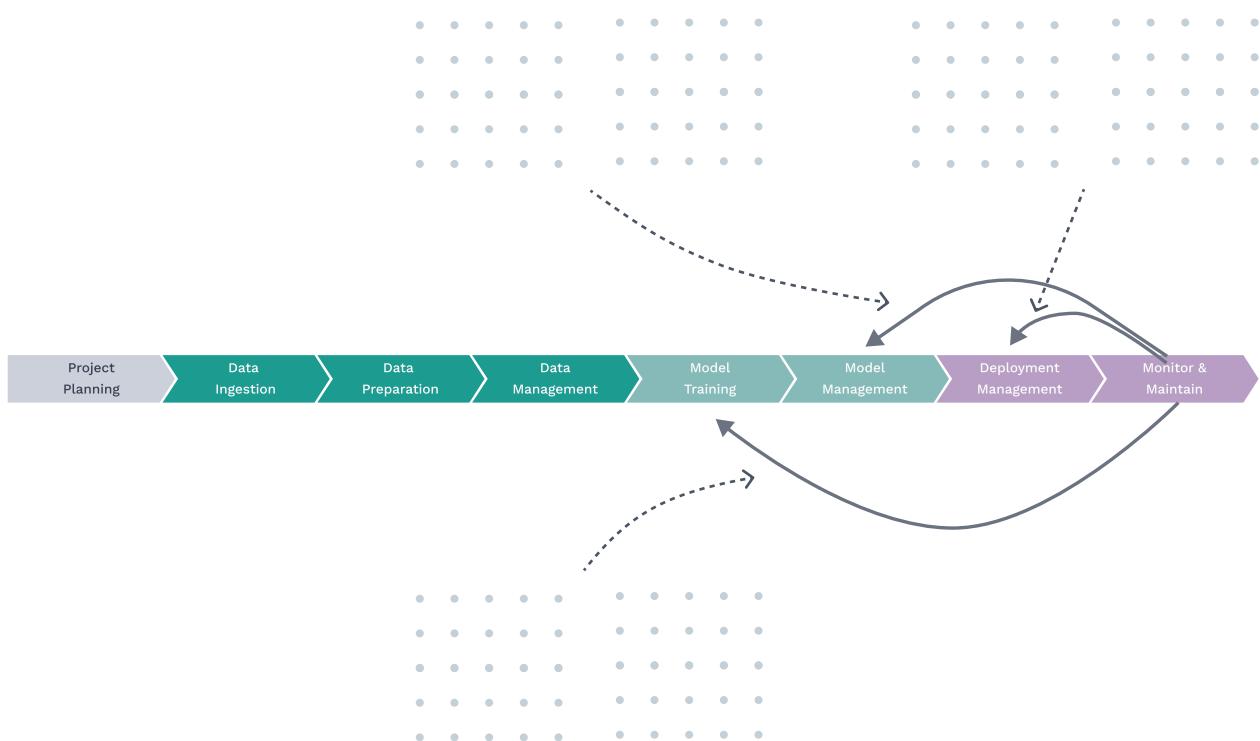
CI/CD

A best practice that is often part of orchestrating a workflow are continuous integration and continuous delivery (CI/CD) pipelines. CI/CD is a set of practices in software development that seek to streamline and automate the process of building, testing, and deploying software.



The long-range feedback loop triggers

Triggers serve as the initial points from which your team revisits an activity in the machine learning lifecycle and iterates on one of the previous phases. Two types of feedback loops exist: long-range and wide-range. In this exercise, attempt to match the triggers listed below with their respective feedback loops in the diagram. Keep in mind that one feedback loop corresponds to two triggers.



1

Too much traffic: Scale up deployment

2

Slow inference time: Reduce model complexity to shrink model size

3

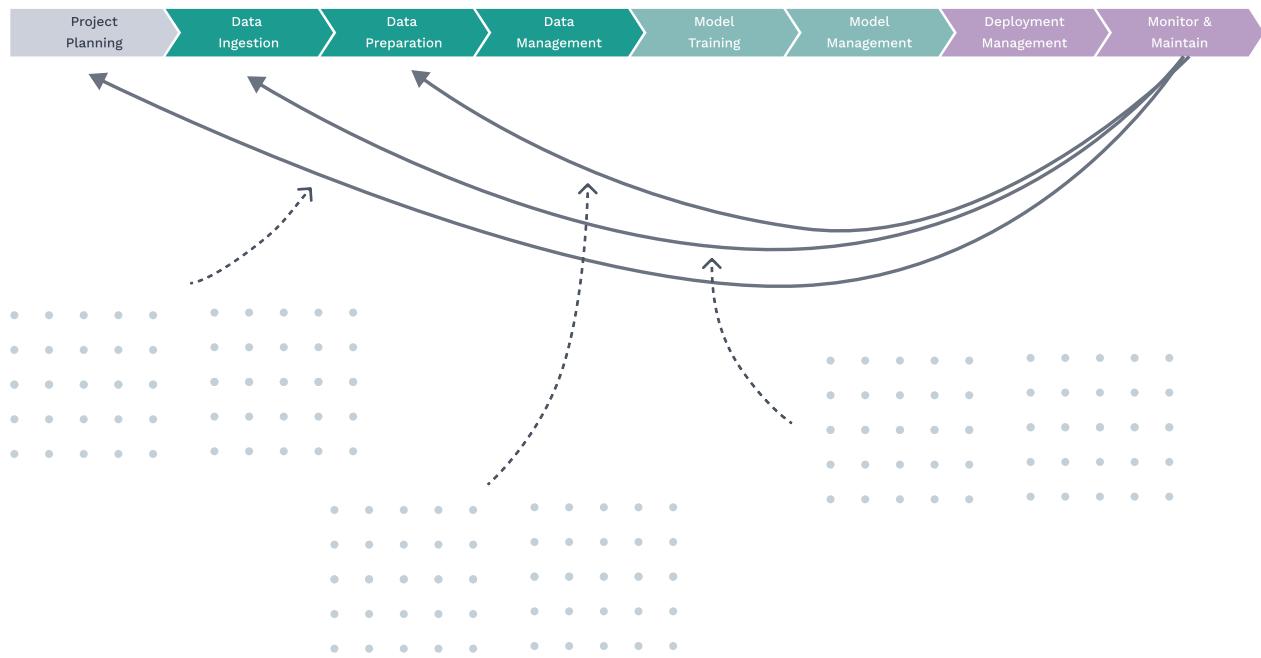
Model biased against certain minorities: Remove ethical features

4

Latest model has worse performance than previous model

The wide-range feedback loop triggers

In this exercise, attempt to match the triggers listed below with their respective feedback loops in the diagram. Keep in mind that the feedback loops provided might not be equally distributed.



1

Data shows consistently different input schema:
Revise data collection strategy

2

Model biased against certain minorities: Revise data collection strategy

3

Correct for data drift by only taking into account more recent data

4

Bad model performance for a specific prediction class: Collect more data and/or improve labeling

5

Model interpretability analysis does not align with expectations: Iterate on data understanding and preparation

6

Model is used differently from the way it is intended to be used than previous model

Practical recommendations

If you're uncertain about what to do next, consider these practical recommendations.

Implement workflow orchestration in a production environment to enhance collaboration, scalability, and a seamless re-deployment process.

Use continuous integration and continuous deployment (CI/CD) pipelines to guarantee adherence to rigorous code standards.

Consider setting up feedback loops and creating backup plans to rollback old models when needed.

Depending on your use case, identify and implement the most suitable model retraining triggers to ensure optimal model performance in the long run.

Self-assessment of comprehension

Take a moment to answer the following questions about the content covered in this module. Keep in mind that there's only one correct answer for each question.

1. Under which circumstances is it reasonable to go back to the Planning Phase after a model has been deployed?

- A) The data shows consistently different input schemes.
- B) The model is used differently from the way it is intended to be used.
- C) The model is negatively biased against certain minorities

2. How does the workflow orchestration tool benefit the ML Lifecycle?

- A) The orchestrator oversees the model's' performance in order to assess their effectiveness in real-world environments.
- B) The orchestrator potentially automates numerous tasks, thereby facilitating easier iterations.
- C) The orchestrator ensures that the entire team stays informed by promptly notifying engineers of any changes in the code base.

3. Which of the following is a feature of a Directed Acyclic Graph (DAG)?

- A) It allows for efficient topological sorting, which means the nodes can be arranged in a linear order such that for every directed edge from node A to node B, A comes before B in the order.
- B) It allows bidirectional edges between nodes, enabling data flow in both directions.
- C) It allows for simultaneous traversal of multiple paths from one node to another, promoting parallel processing.

Self-assessment of comprehension

4. Consider the following scenario: Your team encounters a challenge where the production data consistently shows variations in input structure. Determine the appropriate feedback loop within the ML Lifecycle that should be employed.

- A) Monitor & Maintain -> Data Ingestion
- B) Monitor & Maintain -> Data Management
- C) Monitor & Maintain -> Data Preparation

5. Under which of the following three scenarios is it advisable NOT to use a workflow orchestration tool?

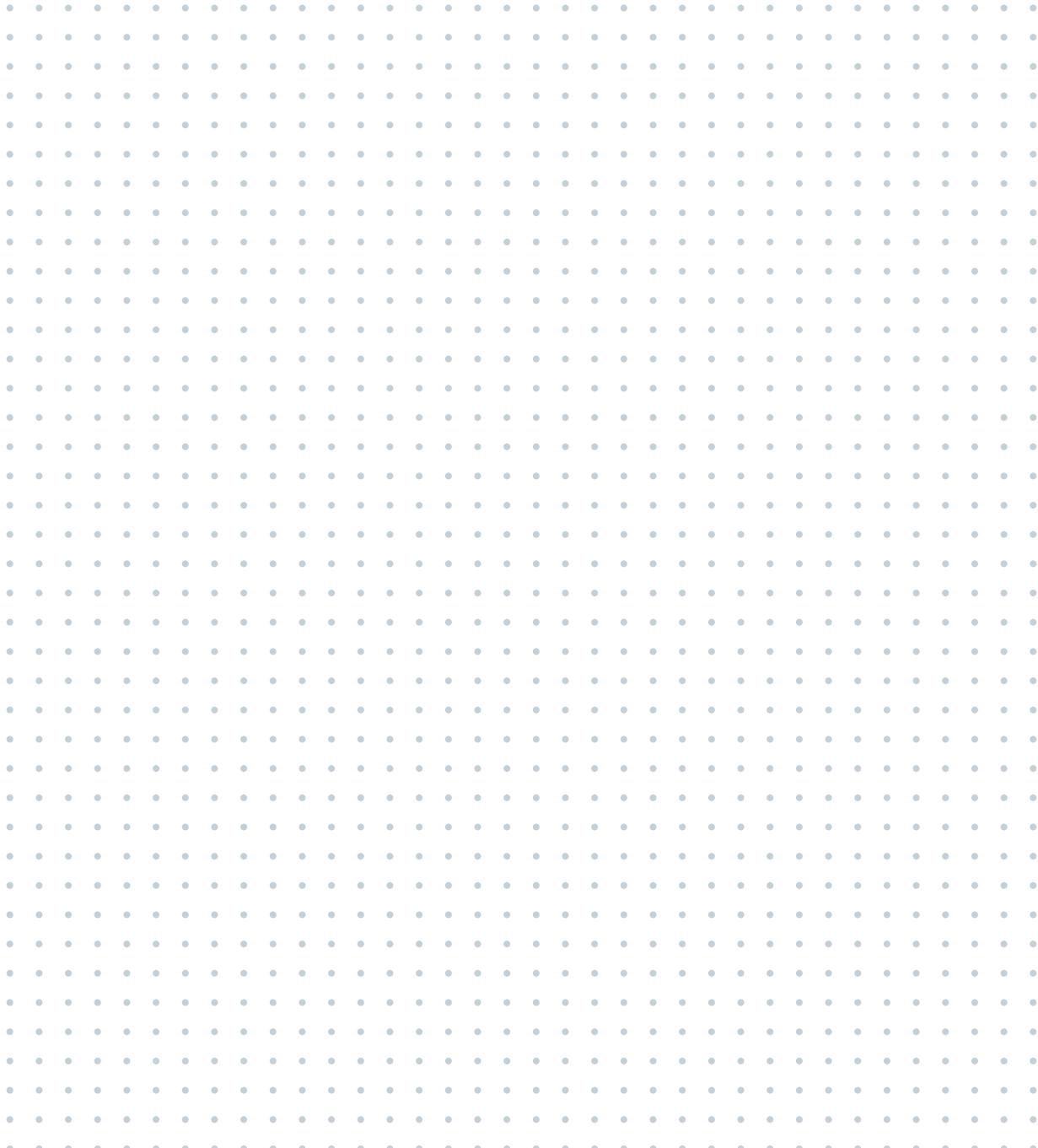
- A) It is advisable not to use a workflow orchestration tool in Machine Learning systems when the system operates in a highly regulated environment with strict compliance requirements.
- B) It is always advisable to use workflow orchestration tools in every phase of the project.
- C) When the phase of the project requires rapid prototyping.

6. How do data scientists benefit from workflow orchestration tools?

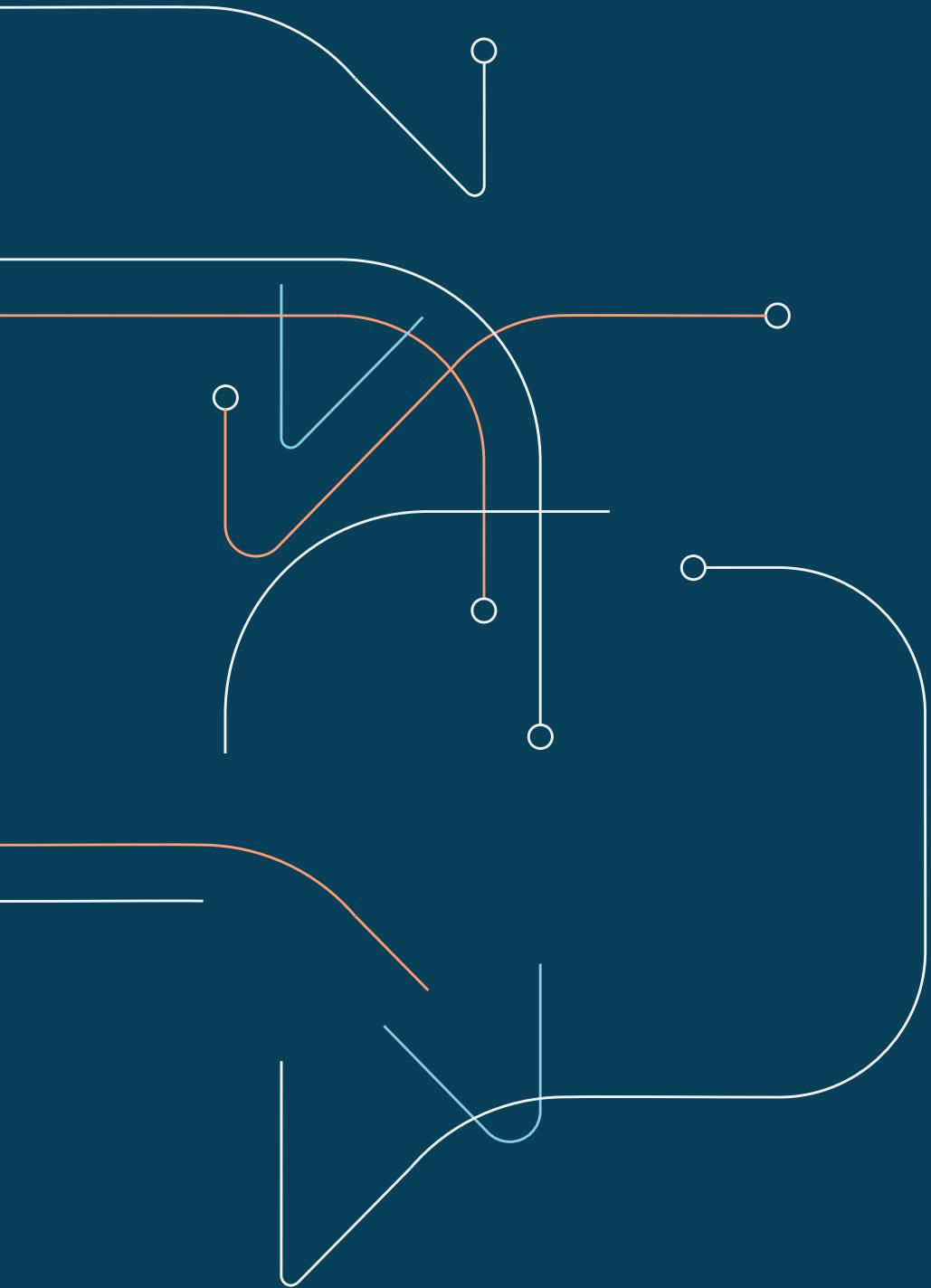
- A) They don't, their job just got automated.
- B) They can save time by avoiding the need to reexecute tasks.
- C) With the availability of data, they can now initiate a workflow to address any business problem effectively.

Module review

Each module begins and ends with this page. At the beginning of the module, we asked you to write down some questions that you would like answered at the end. Once you have written down these questions and worked through the content, take some time to look at the questions again and explain the answers to yourself.

A large grid of grey dots on a white background, intended for handwritten notes. The grid consists of approximately 20 columns and 30 rows of dots, providing a structured area for writing.

07 Credits



Developers

This offering was developed by the appliedAI gGmbH Institute, a non-profit organisation that aims to educate AI professionals in Europe.



Dr. Christian Burkhart

Ownership, Instructional Design, Coordination

Christian Burkhart is a Senior Instructional Designer at appliedAI Institute for Europe gGmbH. His professional career in the area of education spans 15 years. Currently, his work focuses on developing educational products on AI and supporting professionals in their AI education. Before joining appliedAI, he earned his PhD in educational science from the University of Freiburg, Germany.



Omar Eladawy

Subject Matter Expertise Provision, Instruction

Omar Eladawy is an AI Trainer at appliedAI Initiative GmbH. His prior positions include AI Engineer at VoiceLine GmbH and Visiting Student Researcher at the Designing Education Lab at Stanford University. With this unique blend, Omar's experiences bridges Tech, Education and Product. He has been trained in Electrical Engineering and Information Technology.



Alexander Machado

Subject Matter Expertise Provision, Instruction

Alexander works as Head of MLOps Processes at appliedAI Initiative GmbH. He has almost a decade of experience in Data Science, Artificial Intelligence, and Data Engineering at appliedAI, the Max Planck Society, and BMW. His work focused on leading, planning, and developing AI solutions from experimentation to production. He is currently dedicated to innovating technical ML processes that tackle the inherent challenges of ML production systems. These innovations form the foundations of this MLOps online course.



Dr. Jan Willem Kleinrouweler

Subject Matter Expertise Provision

Jan works as Head of MLOps at appliedAI Initiative GmbH. Prior to that, he worked for three years as a Senior Software Engineer for appliedAI and as a Research Scientist at TNO, where he specialized in 5G networks, edge computing and IoT. He received his PhD in distributed interactive systems from Vrije Universiteit Amsterdam, focusing on the optimization of network-based video streaming.

Developers

This offering was developed by the appliedAI gGmbH Institute, a non-profit organisation that aims to educate AI professionals in Europe.



Rocío Estrada

Visual Design

Rocío Estrada is an Instructional Designer at appliedAI Institute for Europe gGmbH, bringing 4 years of experience in crafting solutions for recruitment and training programs. While focused on creating educational offerings, her skill set extends to encompass visual design and video editing, enhancing the learning experience.



Lamin Yannick Rubas

User Testing

Lamin Rubas is an AI Trainer at appliedAI Initiative GmbH. His work surrounds the development and delivery of AI education products. He previously worked in the Learning Department at Boston Consulting Group and has a background in Management and Computer Science.



Kanaat Bektas

Technical Integration, Video Recording, Video Editing

Kanaat Bektas is an Senior eLearning Specialist at appliedAI Initiative GmbH, having 6-years of experience in the fields of eLearning, LMSs, Instructional Design, and Video Production. With a background in Computer Education and Education Technology, Kanaat brings knowledge and expertise to the team, dedicated to enhancing the online education and learning experiences.



Anna-Maria Krenz

Marketing

Anna-Maria Krenz is responsible for the Marketing at the appliedAI Institute for Europe gGmbH. She has a background in communication, product and brand management. She currently holds the position of Principal Communication and Brand Manager at the appliedAI Institute for Europe. With previous six years experience at Amazon, she brings expertise in areas such as customer centricity, product strategy, marketing automation.

About appliedAI

The appliedAI Institute for Europe aims to strengthen the European AI ecosystem, develop knowledge around AI, provide trusted AI tools, and create educational and interactive formats around high-quality AI content.

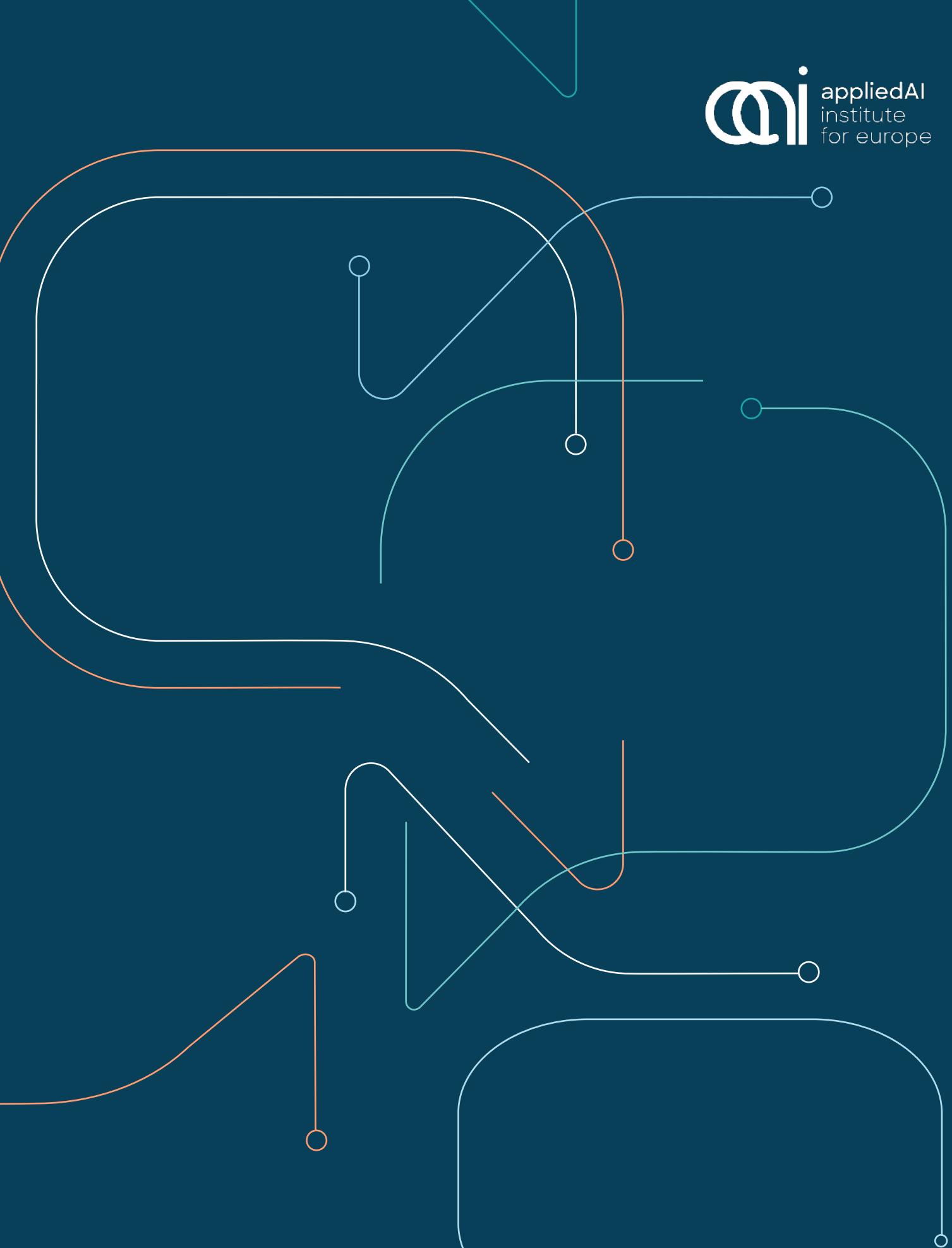
As a non-profit subsidiary of the appliedAI Initiative, the institute was founded in Munich in 2022. The appliedAI Initiative itself is a joint venture of UnternehmerTUM and IPAI. The institute is managed by Dr. Andreas Liebl and Dr. Frauke Goll.

The appliedAI Institute for Europe focuses on the people in Europe. It pursues the vision of shaping a common AI community and providing high-quality content in the age of AI for the entire society. By promoting trustworthy AI, the Institute accelerates the application of this technology and strengthens trust in AI solutions.

With a focus on knowledge development and the provision of trusted AI tools, the appliedAI Institute for Europe provides a valuable resource for companies, organizations, and individuals looking to expand their knowledge and skills in AI. Through educational and interaction formats, the Institute enables an intensive exchange of expertise and fosters collaboration between stakeholders from different fields.

The appliedAI Institute for Europe invites companies, organizations, startups, and AI enthusiasts to benefit from the Institute's diverse offerings and resources.

For more information, please visit www.appliedai-institute.de.



appliedAI Institute for Europe
Lichtenbergstr. 6
85748 Garching
Germany
www.appliedai-institute.de