Reaching for the MLOps level 1

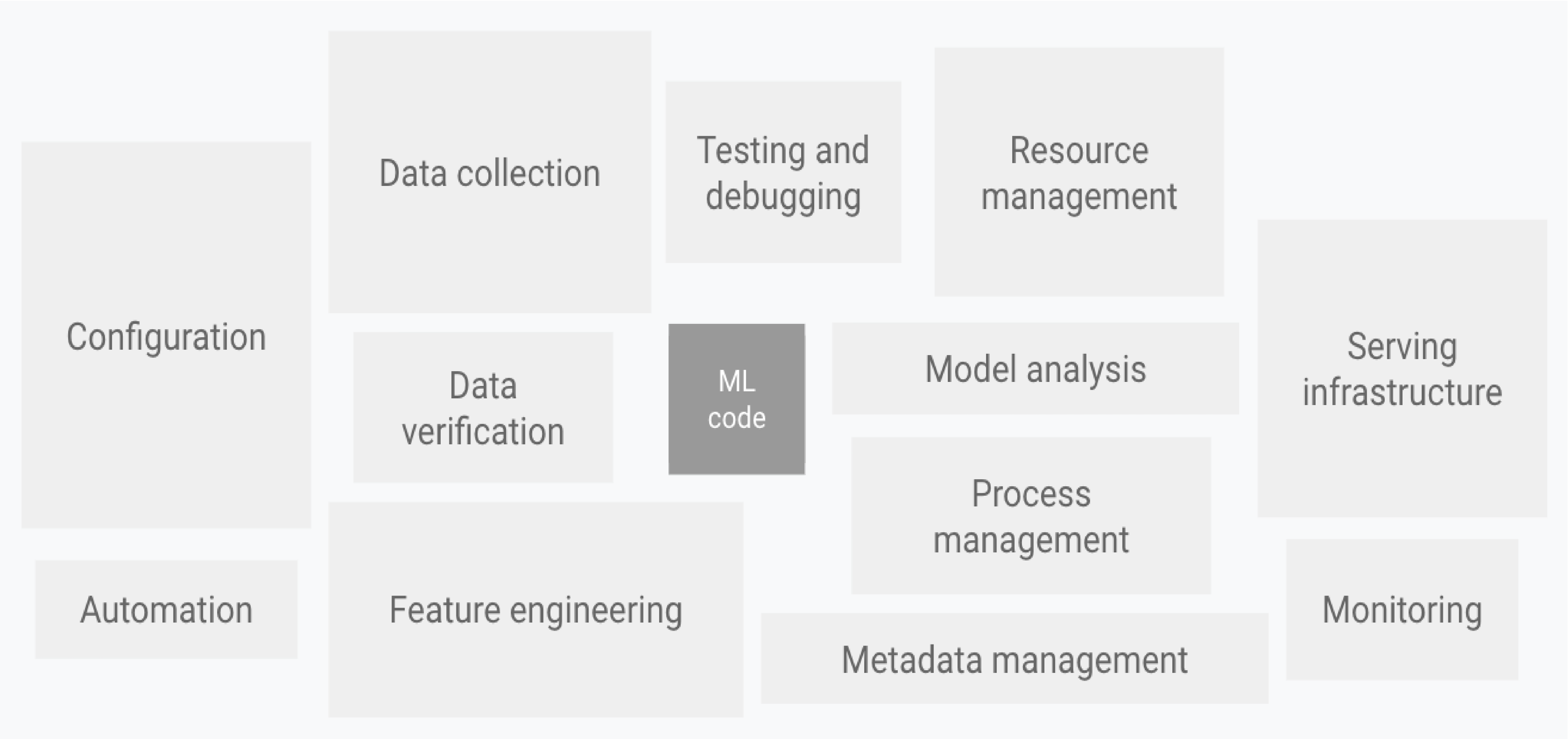
Understanding and implementing MLOps into practice

Machine learning

The scientific community, as well as statistical community, focuses on the development of ML models, not in producing ready-made machine learning products.

That is the reason why so many ML projects fail (based on research).

Focus should be on building ML products, not only on developing and building ML models.



Configuration

Data collection

Testing and debugging

Resource management

Serving infrastructure

Data verification

ML code

Model analysis

Process management

Automation

Feature engineering

Metadata management

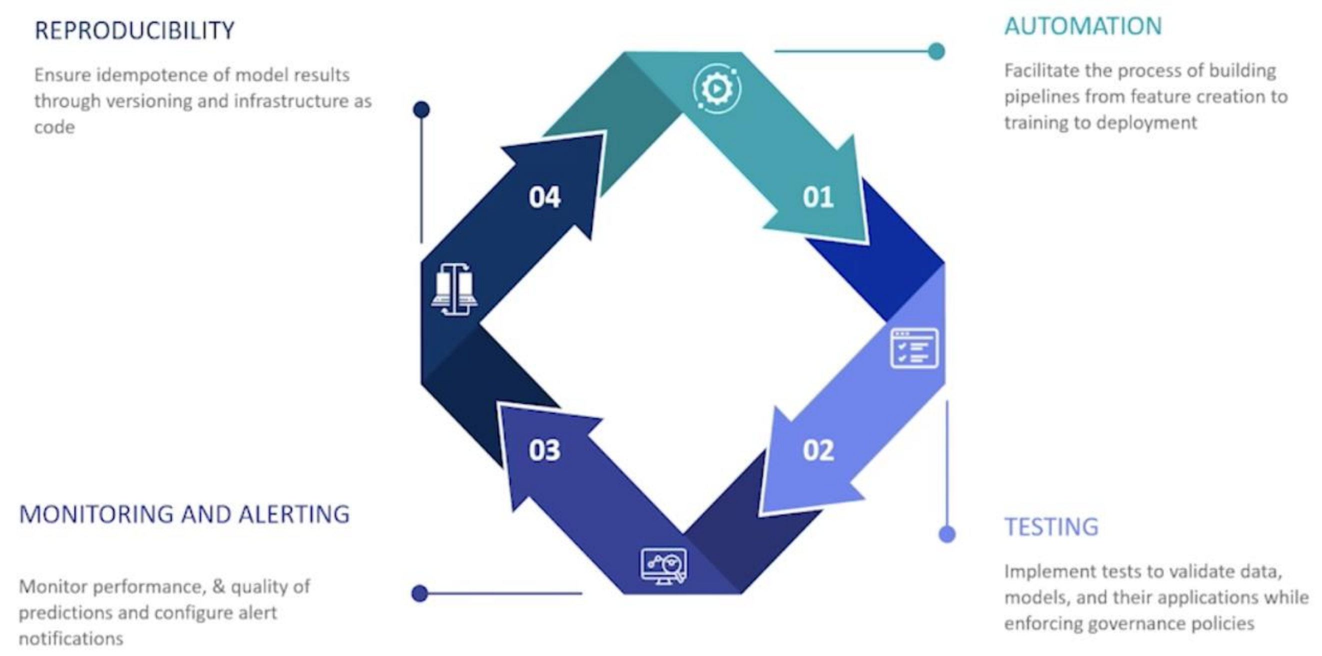
Monitoring

MLOps

- Collection of principles and components that implement those principles
  - Sometimes MLOps as a concept can seem a little “fuzzy” and everyone has a slightly different understanding and description of it.
- Why do we need it?
  - That we could implement necessary quality dimensions (or responsible ML?)
  - Kind of a “horror picture” is that statistics would be based on predictions produced by machine learning models, about which we know nothing
- MLOps Includes also a standardized process (or workflow as they call it) and roles
- ML model is just one component among many other components
  - Actually, model is not a component in MLOps… (it’s one the main artifacts)

Principles

- Versioning
  - Data, code and model
- Automation
- Reproducibility
- Monitoring
- Deployment (CI/CD)



AUTOMATION

Facilitate the process of building pipelines from feature creation to training to deployment

TESTING

Implement tests to validate data, models, and their applications while enforcing governance policies

MONITORING AND ALERTING

Monitor performance, & quality of predictions and configure alert notifications

REPRODUCIBILITY

Ensure idempotence of model results through versioning and infrastructure as code

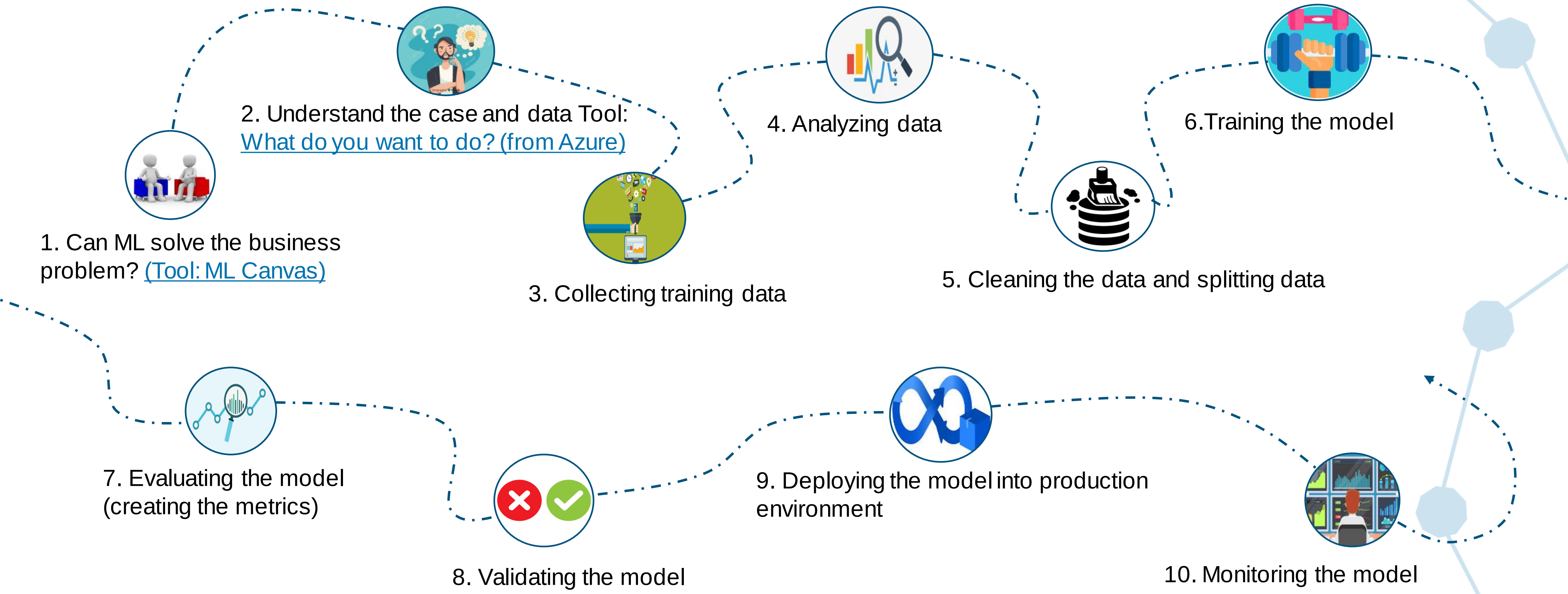
Reproducibility

- Every phase (data processing, model training, model deployment) will produce identical results with the same input
- In other words: you can **repeatedly run your algorithm on certain datasets and obtain the same results**
- Reproducibility is a major principle of the **scientific method** (predictions are produced by scientific methods)
- The lack of reproducibility (and transparency) undermines predictions scientific value
- Reproducibility creates trust and credibility with the ML product
- Versioning is one of the key components behind reproducibility

Automation

- The level of automation (data, model, code pipelines) determines the maturity of the ML process
- From manual process (all the steps are executed manually) to full automation (three levels, from manual data science process to full automated ML process)
- Three ”official” levels:
    - Manual process (Google Cloud’s level 0). Every step in ML process is performed manually.
    - ML pipeline automation (Google’s Level 1). *Continuous Training* and data validation/model monitoring.
    - CI/CD pipeline automation (Google’s Level 2). Building, testing and deploying data and model automatically.

ML process/workflow



1. Can ML solve the business problem? (Tool: ML Canvas)

2. Understand the case and data Tool: What do you want to do? (from Azure)

3. Collecting training data

4. Analyzing data

5. Cleaning the data and splitting data

6.Training the model

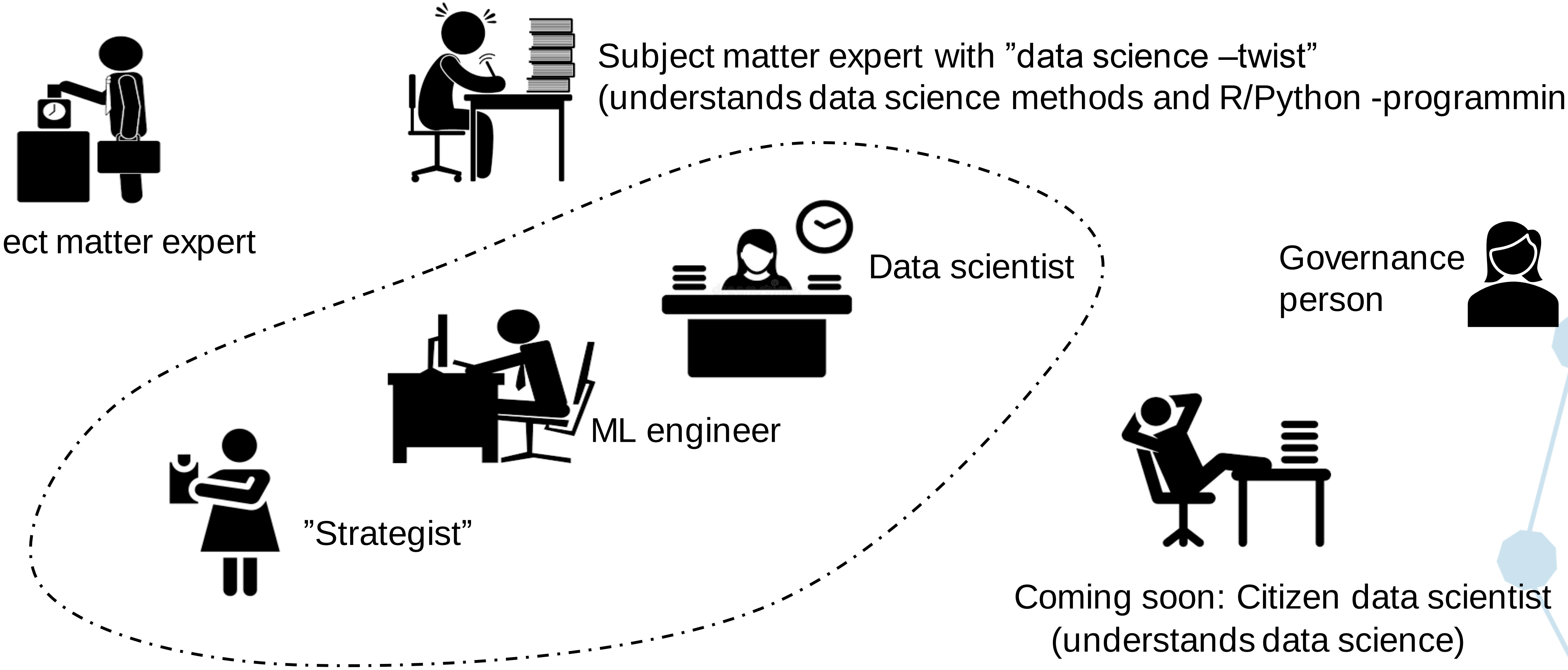
7. Evaluating the model (creating the metrics)

8. Validating the model

9. Deploying the model into production environment

10. Monitoring the model

Roles



Subject matter expert

Subject matter expert with ”data science –twist”
(understands data science methods and R/Python -programming)

Data scientist

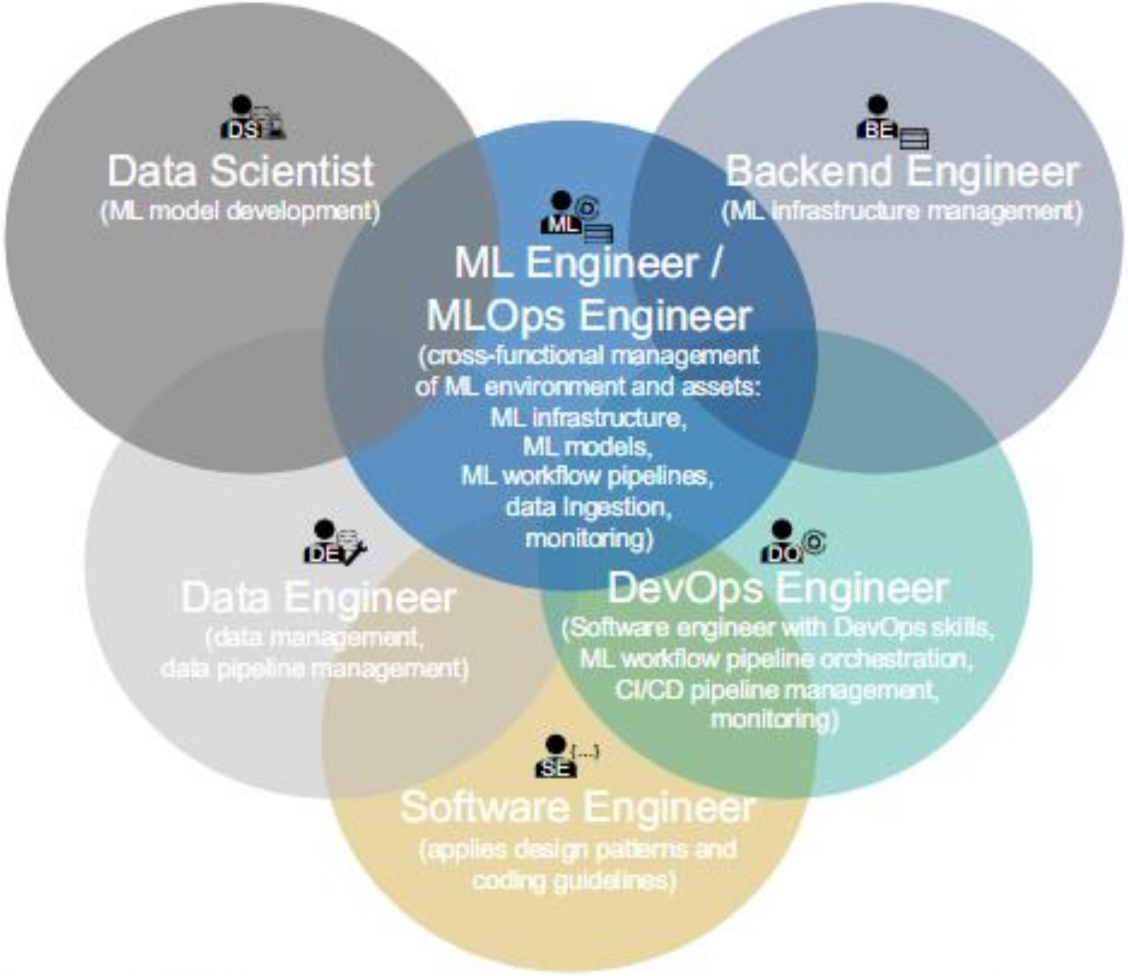
Governance
person

ML engineer

”Strategist”

Coming soon: Citizen data scientist
(understands data science)

…or it can look like this!



Data Scientist
(ML model development)

Backend Engineer
(ML infrastructure management)

ML Engineer /
MLOps Engineer
(cross-functional management
of ML environment and assets:
ML infrastructure,
ML models,
ML workflow pipelines,
data Ingestion,
monitoring)

Data Engineer
(data management,
data pipeline management)

DevOps Engineer
(Software engineer with DevOps skills,
ML workflow pipeline orchestration,
CI/CD pipeline management,
monitoring)

Software Engineer
(applies design patterns and
coding guidelines)

Figure 3. Roles and their intersections contributing to the MLOps paradigm

Monitoring

- Monitoring accuracy/quality of models prediction (also the quality of training data should be monitored)
- In fully automated MLOps workflow monitor triggers the CT (continuous training)
- But, when the process is not fully automated the monitor can give an alarm to a governance person
- There are open source –libraries and off-the-shelf –products (I also presume, that in some off-the-shelf –ML platforms, monitor components are implemented into the platform)

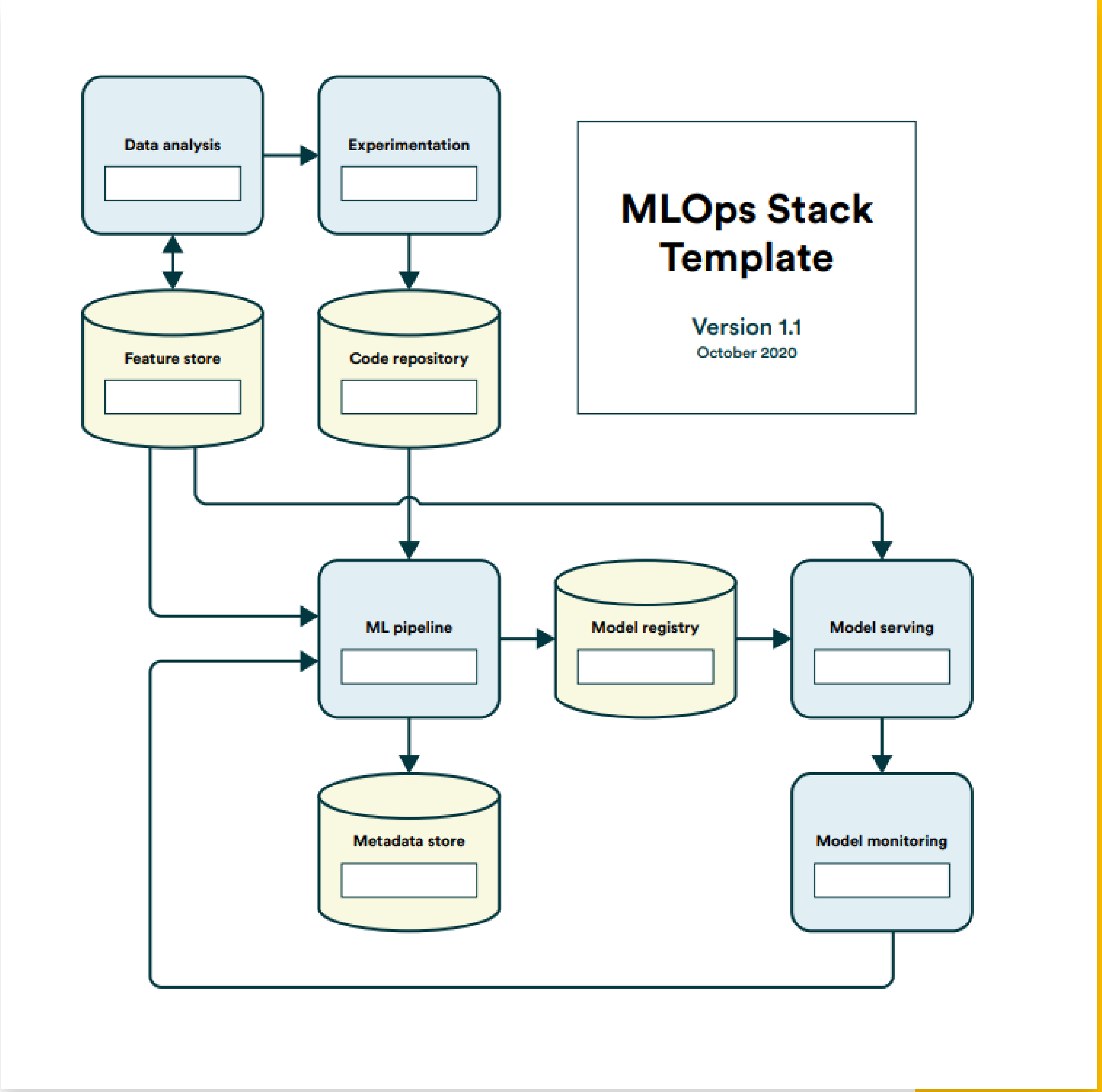
Components

- Technical components implement MLOps principles
  - Source control/source code repository
  - Feature store
  - CI/CD component
  - Model serving
  - Model registery
  - Monitoring component
  - Pipeline/workflow orchestrator
  - Metadata store

MLOps platform?

- Definition of MLOps platform is even ”fuzzier” than the definition of MLOps
  - Sometimes difficult to decribe the 1:1 connections between ML platforms and principles of MLOps –theory
- ML Platforms can be more like infrastructure platforms or data management or business intelligence platforms (example: Pyramid Analytics) or…
- Off-the-shelf machine learning platform versus in-house solution, using open-source tools and libraries on existing cloud service provides platform
- Some platforms go more to Citizen data scientist –direction!

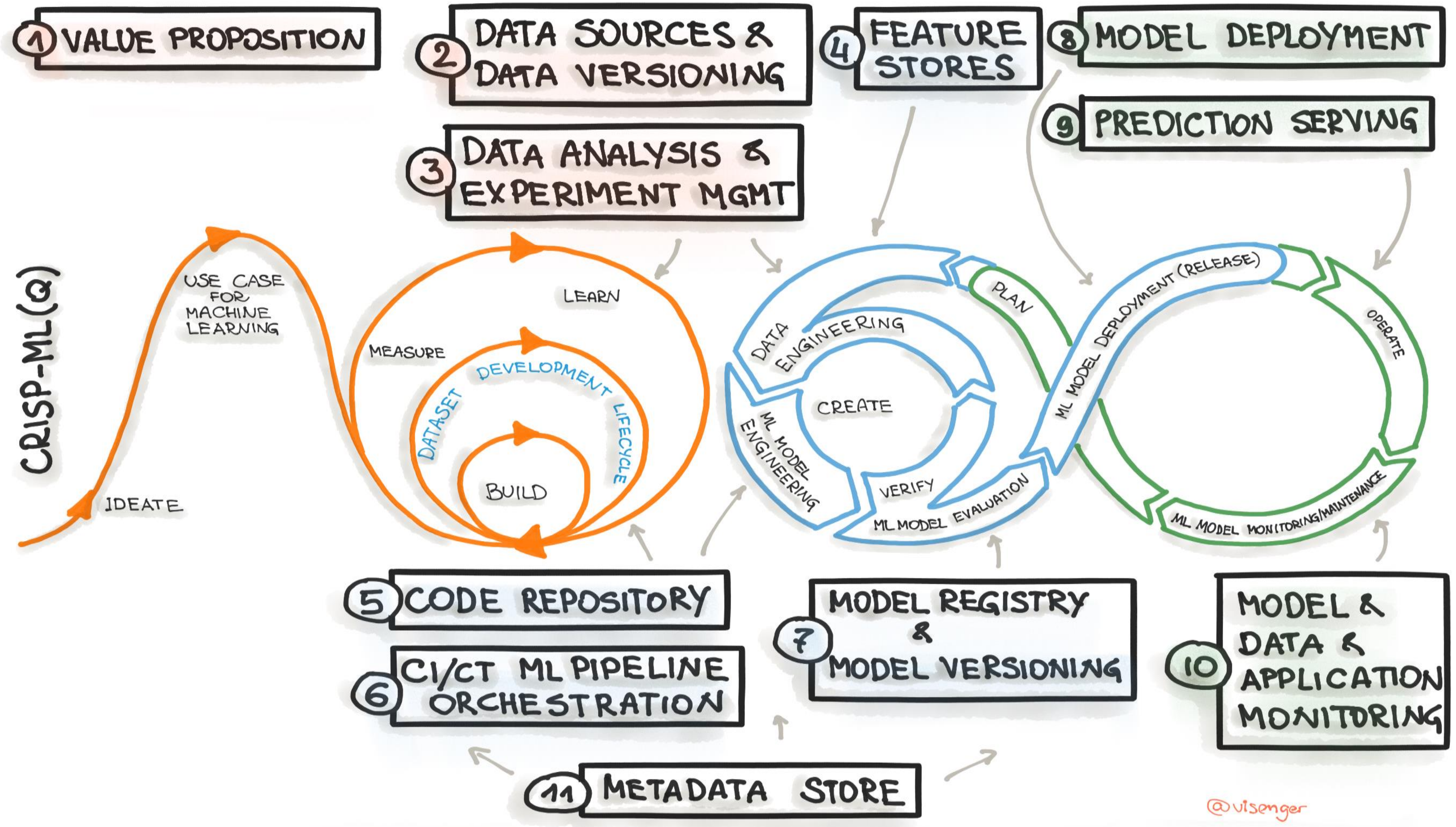
An example: MLOps platform (or Stack!) from Valohai



MLOps Stack?

MLOps Stack Canvas, Online Whiteboard for Visual Collaboration (miro.com)

MLOPS STACK



1 VALUE PROPOSITION
2 DATA SOURCES & DATA VERSIONING
3 DATA ANALYSIS & EXPERIMENT MGMT
4 FEATURE STORES
8 MODEL DEPLOYMENT
9 PREDICTION SERVING
CRISP-ML(Q)
IDEATE
USE CASE FOR MACHINE LEARNING
MEASURE
LEARN
DATASET DEVELOPMENT LIFECYCLE
BUILD
DATA ENGINEERING
CREATE
ML MODEL ENGINEERING
VERIFY
ML MODEL EVALUATION
PLAN
ML MODEL DEPLOYMENT (RELEASE)
OPERATE
ML MODEL MONITORING/MAINTENANCE
5 CODE REPOSITORY
6 CI/CT ML PIPELINE ORCHESTRATION
7 MODEL REGISTRY & MODEL VERSIONING
10 MODEL & DATA & APPLICATION MONITORING
11 METADATA STORE
@visenger

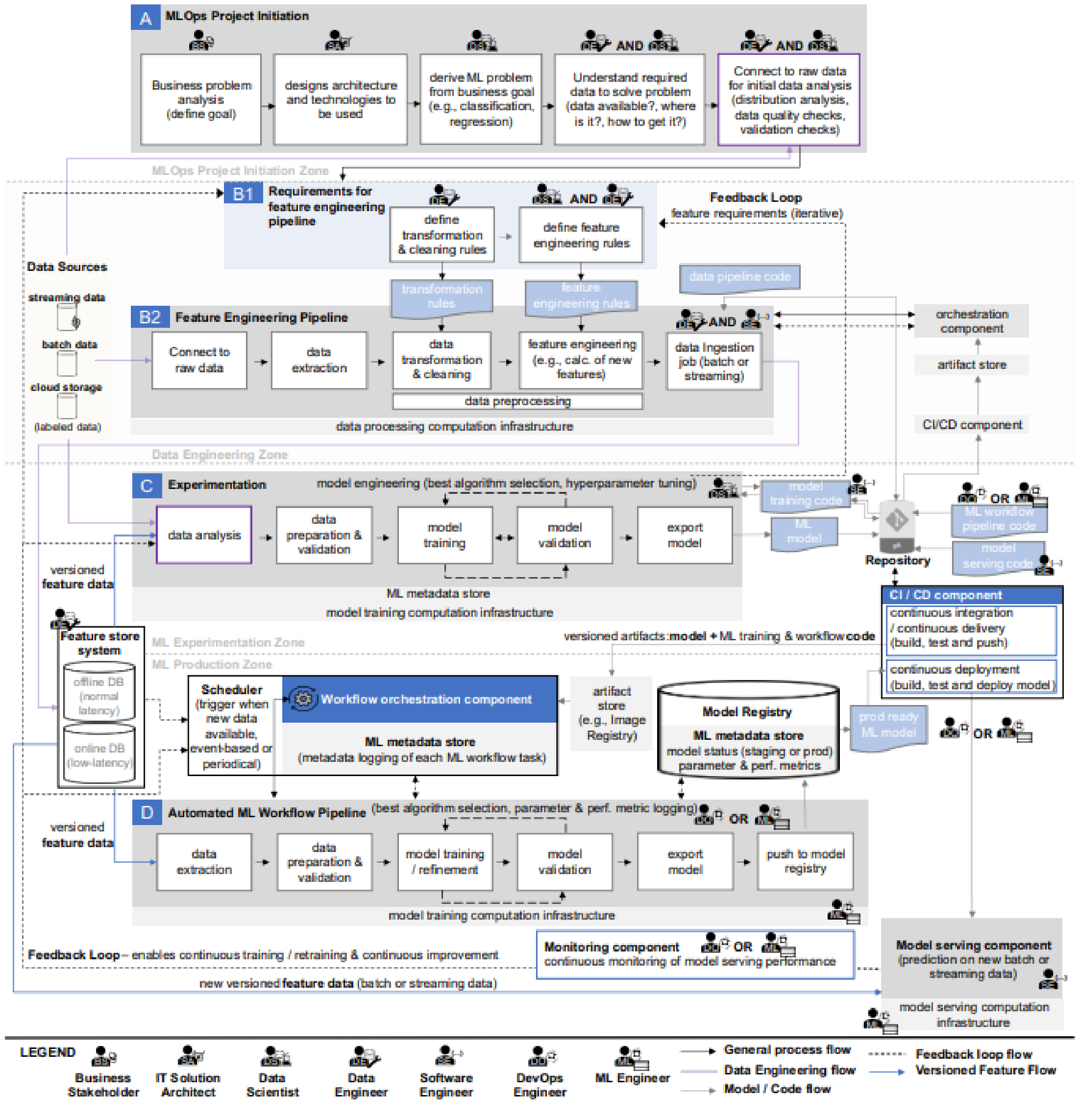
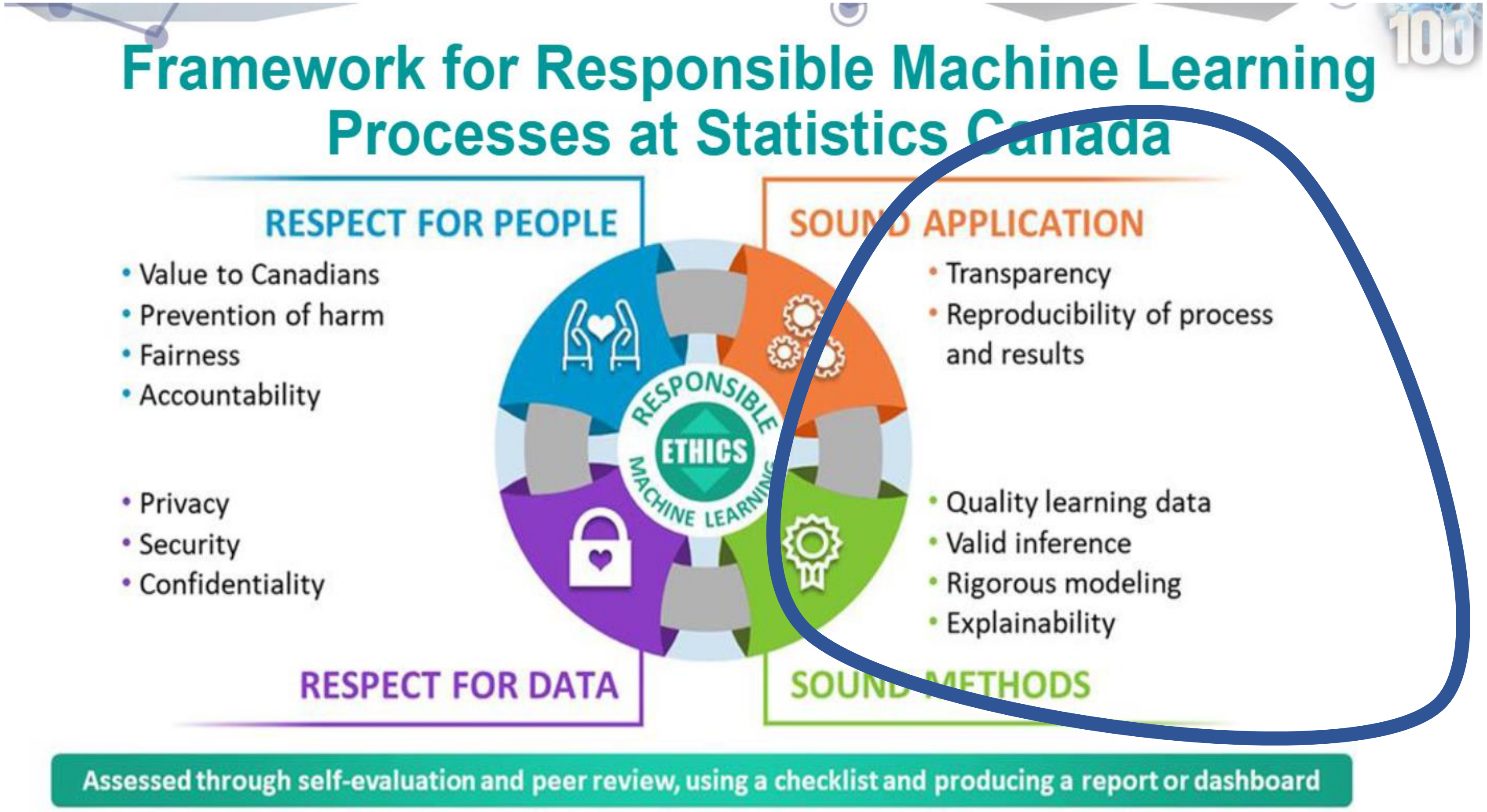

A
MLOps Project Initiation
BS
SA
DS
DE AND DS
DE AND DS
Business problem analysis (define goal)
designs architecture and technologies to be used
derive ML problem from business goal (e.g., classification, regression)
Understand required data to solve problem (data available?, where is it?, how to get it?)
Connect to raw data for initial data analysis (distribution analysis, data quality checks, validation checks)
MLOps Project Initiation Zone
B1
Requirements for feature engineering pipeline
define transformation & cleaning rules
DS AND DE
define feature engineering rules
Feedback Loop
feature requirements (iterative)
Data Sources
streaming data
batch data
cloud storage
(labeled data)
transformation rules
feature engineering rules
data pipeline code
B2
Feature Engineering Pipeline
Connect to raw data
data extraction
data transformation & cleaning
feature engineering (e.g., calc. of new features)
DE AND SE
data ingestion job (batch or streaming)
data preprocessing
data processing computation infrastructure
orchestration component
artifact store
CI/CD component
Data Engineering Zone
C
Experimentation
model engineering (best algorithm selection, hyperparameter tuning)
data analysis
data preparation & validation
model training
model validation
export model
model training code
ML model
DO OR ML
ML workflow pipeline code
model serving code
Repository
ML metadata store
model training computation infrastructure
versioned feature data
Feature store system
offline DB (normal latency)
online DB (low-latency)
CI / CD component
continuous integration / continuous delivery (build, test and push)
continuous deployment (build, test and deploy model)
ML Experimentation Zone
ML Production Zone
versioned artifacts: model + ML training & workflow code
Scheduler (trigger when new data available, event-based or periodical)
Workflow orchestration component
ML metadata store (metadata logging of each ML workflow task)
artifact store (e.g., Image Registry)
Model Registry
ML metadata store
model status (staging or prod) parameter & perf. metrics
prod ready ML model
DO OR ML
D
Automated ML Workflow Pipeline (best algorithm selection, parameter & perf. metric logging)
DO OR ML
versioned feature data
data extraction
data preparation & validation
model training / refinement
model validation
export model
push to model registry
model training computation infrastructure
Feedback Loop – enables continuous training / retraining & continuous improvement
Monitoring component
DO OR ML
continuous monitoring of model serving performance
Model serving component (prediction on new batch or streaming data)
model serving computation infrastructure
new versioned feature data (batch or streaming data)
LEGEND
Business Stakeholder
IT Solution Architect
Data Scientist
Data Engineer
Software Engineer
DevOps Engineer
ML Engineer
General process flow
Data Engineering flow
Model / Code flow
Feedback loop flow
Versioned Feature Flow

Figure 4. End-to-end MLOps architecture and workflow with functional components and roles

Responsible ML (Statistics Canada) and MLOps



Framework for Responsible Machine Learning Processes at Statistics Canada

RESPECT FOR PEOPLE

- Value to Canadians
- Prevention of harm
- Fairness
- Accountability

SOUND APPLICATION

- Transparency
- Reproducibility of process and results

RESPECT FOR DATA

- Privacy
- Security
- Confidentiality

SOUND METHODS

- Quality learning data
- Valid inference
- Rigorous modeling
- Explainability

Assessed through self-evaluation and peer review, using a checklist and producing a report or dashboard

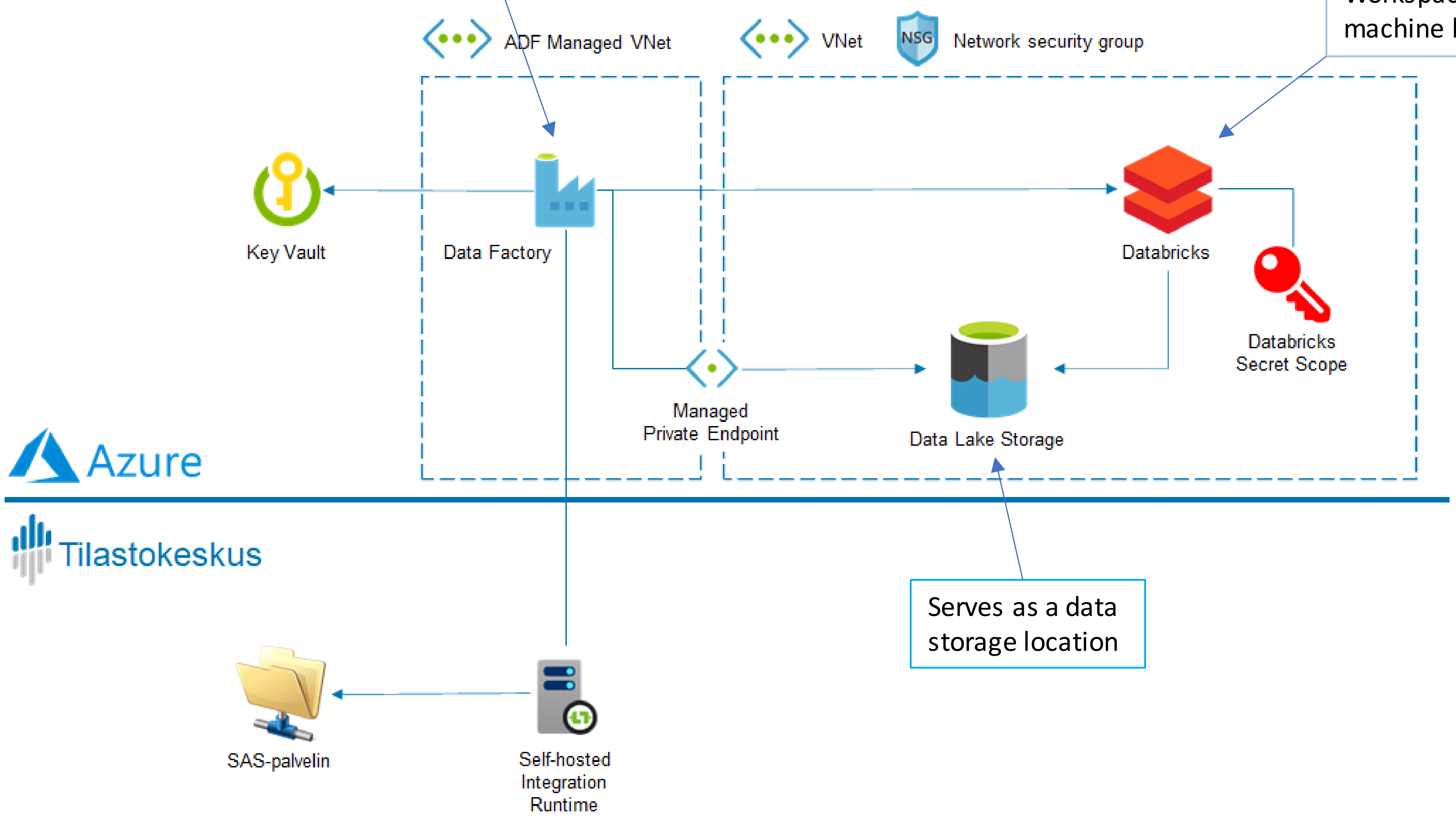
What we have done at Statistics Finland?

- We have built a ML platform (in-house)
- Deployed two ML models into production at that platform
- We have also studied the theory of MLOps…
- We are planning to use Model Cards as Model’s metadata description (improving the transparency and explainability)
- From MLOps (automation) maturity point of view, we are still at the level 0…

Model Card Toolkit | Responsible AI Toolkit | TensorFlow

GitHub - tensorflow/model-card-toolkit: a tool that leverages rich metadata and lineage information in MLMD to build a model card

Our ”in-house” -platform



We use Data Factory -instance to move data between cloud and on-prem environments and launch prediction runs.

Workspace for developing machine learning models

Serves as a data storage location

MLOps principles at Statistics Finland

- Versioning
  - models are versioned in model registry
  - code (no versioning)
  - data (not versioned although it’s possible with Delta Lake)
- Automation
  - now at level 0 (+)
- Reproducibility (missing parts…)
- Monitoring (we are looking for a solution…)
- Deployment (CI/CD) (when predictions are requested, the register is automatically searched, but nothing else…)
- Many ”development lines”:
  - Implementing ML to statistical business processes (automating manual classification)
  - The more machine learning solutions there are in production, more principles must be implemented into our MLOps platform

Future (global development)

- Citizen data scientist
  - Is part of the low code/no-code/democratization – development
  - All the big consulting companies (Gartner, Deloitte etc., see it coming)
- Let’s be ready for this development as well, but first MLOps to implement features of Responsible ML.