NLP ML-Ops

LIAD MAGEN

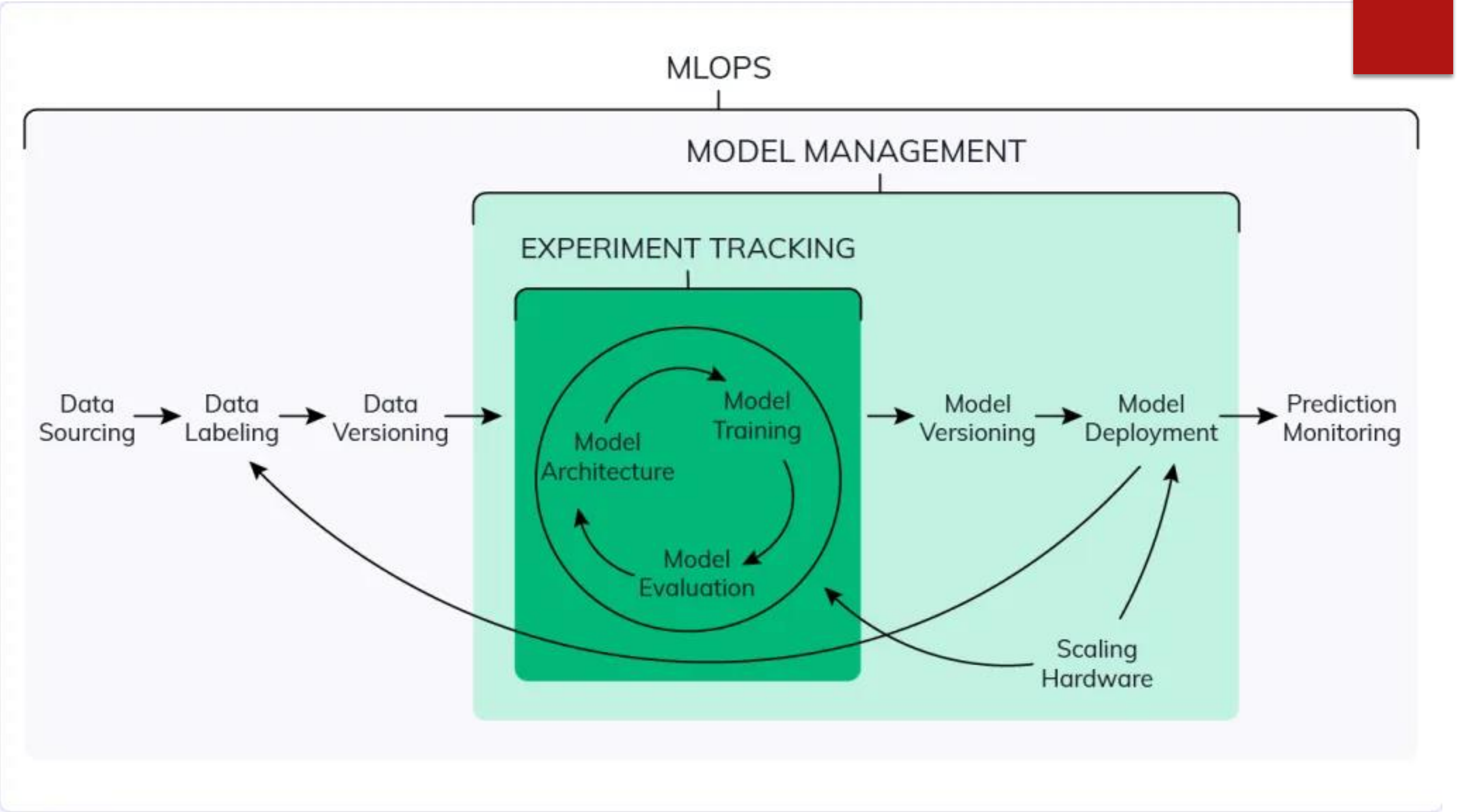
```
irror_mod = modifier_ob.
 mirror object to mirror
irror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
 operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
 operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

selection at the end -add
_ob.select= 1
ier_ob.select=1
ntext.scene.objects.activ
("Selected" + str(modifier
irror_ob.select = 0
 bpy.context.selected_obj
ata.objects[one.name].se

int("please select exactl

-- OPERATOR CLASSES ----

types.Operator):
 X mirror to the selected
ject.mirror_mirror_x"
rror X"

ontext):
ext.active_object is not
```

ML is Software Development

What are ML-Ops?

ML-OPS

- Set of communication fostering practices
- Helpful Tools for Machine Learning Operations
  - Experiments tracking
  - Experiments reproducing
  - CI/CD
  - Data processing on scale
  - Hyperparameters tuning
  - Model Interpretation
  - Model packing / serving

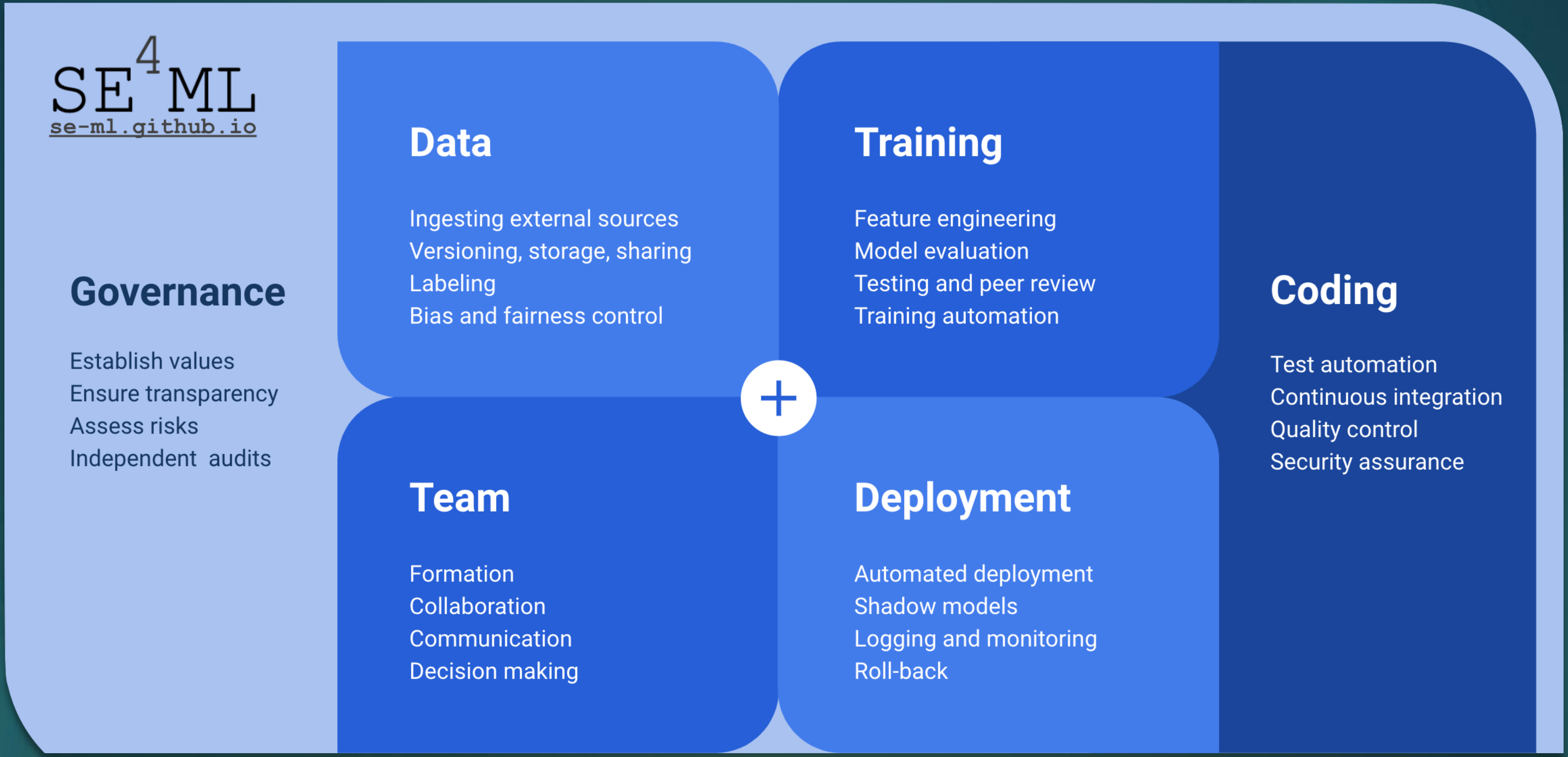

MLOPS
MODEL MANAGEMENT
EXPERIMENT TRACKING
Data Sourcing
Data Labeling
Data Versioning
Model Architecture
Model Training
Model Evaluation
Model Versioning
Model Deployment
Prediction Monitoring
Scaling Hardware

Some Tools Examples

- Labeling
- Experiment Tracking
  - MLFlow
  - ClearML
- Data Versioning
  - DVC
- Hyperparameter Optimization
  - Optuna
- Pipeline orchestration
  - Airflow
- Model Deployment
  - BentoML
  - Seldon
- And many more...

Exhaustive list:

HTTPS://ABOUT.MLREEF.COM/BLOG/GLOBAL-MLOPS-AND-ML-TOOLS-LANDSCAPE



SE^{4}ML
se-ml.github.io

Governance

Establish values
Ensure transparency
Assess risks
Independent audits

Data

Ingesting external sources
Versioning, storage, sharing
Labeling
Bias and fairness control

Training

Feature engineering
Model evaluation
Testing and peer review
Training automation

Team

Formation
Collaboration
Communication
Decision making

Deployment

Automated deployment
Shadow models
Logging and monitoring
Roll-back

Coding

Test automation
Continuous integration
Quality control
Security assurance

Practical Advice

ENGINEERING BEST PRACTICES FOR MACHINE LEARNING
HTTPS://SE-ML.GITHUB.IO/PRACTICES/

Data

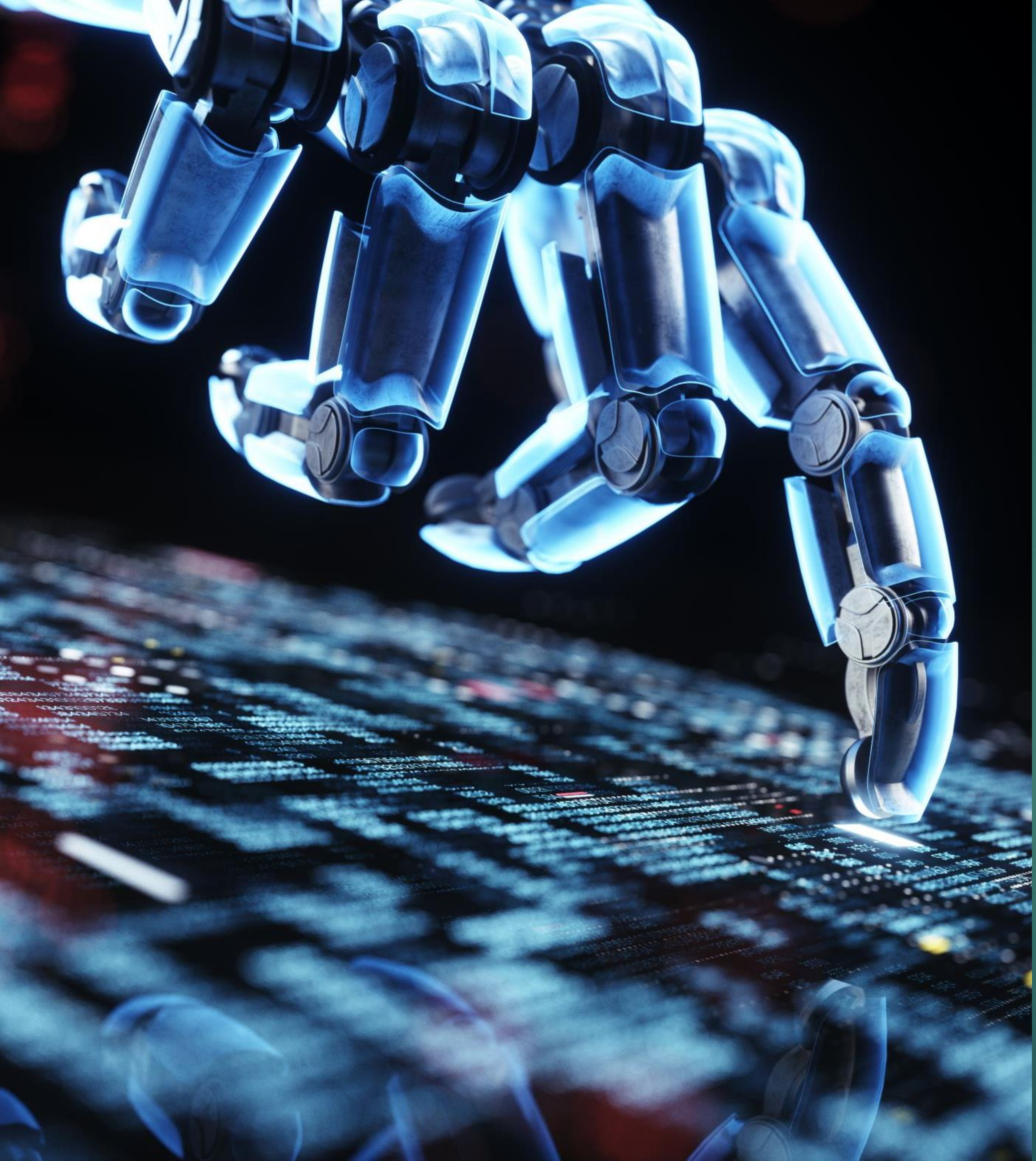
- Good labeling is the most important thing
- Use Versioning Tools
  - DVC
  - Pachyderm (K8S)
- Be able to reproduce preprocessing results
- Automatic Data Validation (Great Expectations)
- Data Engineering Convention (Kedro – QuantumBlack):
  - Raw
  - Intermediate
  - Primary
  - Feature
  - Model Input
  - Models
  - Model output
  - Reporting

Team

- Enable anyone to reproduce your results
  - Also true for you, *k* months from now
- Read papers and share knowledge
- Discuss
  - Brainstorm
  - The best ideas come when you express them out loud

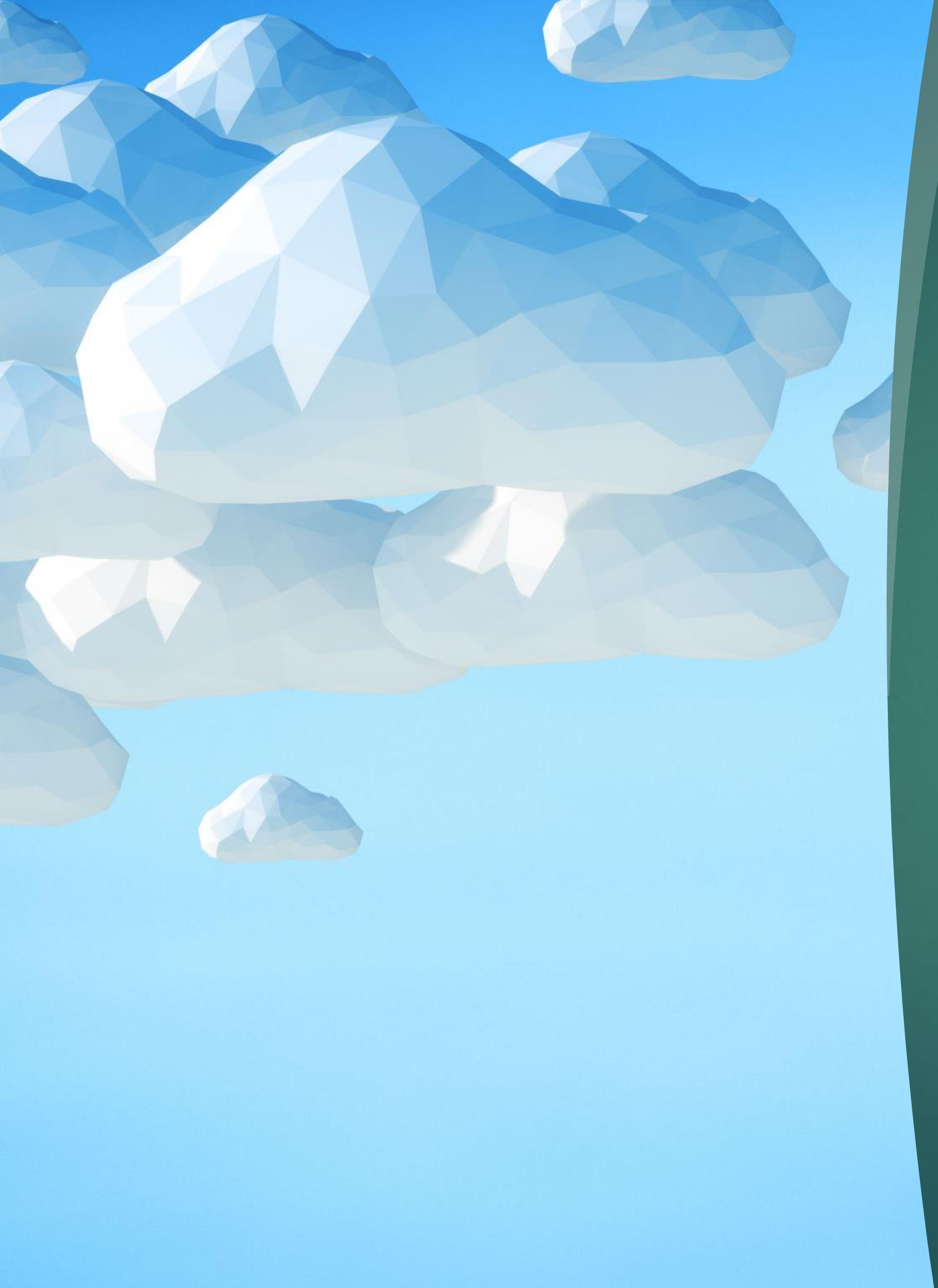
Training

- Start small (few samples)
  - Ensure you can overfit
- Track experiments
- Separate configuration from code
  - e.g. YAML files
- Commit before you execute (!)
- Handle errors
- Use TensorBoard
  - Essential for debugging



Deployment

- Monitor performance
- Be able to explain bad predictions
  - - and good ones too!
- Aspire to automate, but not in any cost



Cloud Automation

- Model deployment
  - Requires more expertise than before
- Model comparison
  - Evaluation metrics
  - Interpretability
  - Performance (inference speed)
- Shadow Model
  - Compete against current model

Never Stop Learning