Machine Learning Operations for Accelerator Control

Tia Miceli (Lead - Accelerator AI/ML Group, Accelerator Controls Department, FNAL)
EPICS Collaboration Meeting
24-28 April 2023

Why you need a sustainable way of developing, deploying, monitoring, and servicing ML applications (for accelerators)

- The only person that knew anything about application / model / code leaves
- The “reproducibility problem” in deep learning
- Life-cycle handling: is it still doing the right thing? If not, what does an accelerator operator do at 3 A.M.?

Why you need a sustainable way of developing, deploying, monitoring, and servicing ML applications (for accelerators)

- The only person that knew anything about application / model / code leaves ⇒ need self-documenting procedures
- The “reproducibility problem” in deep learning ⇒ need advanced and automated “bookkeeping”
- Life-cycle handling: is it still doing the right thing? If not, what does an accelerator operator do at 3 A.M.? ⇒ need to automate common updates

“The Reproducibility Problem” (in AI / ML)

- “I am able to train a model once, but I / someone else can’t reproduce the same model weights again.”

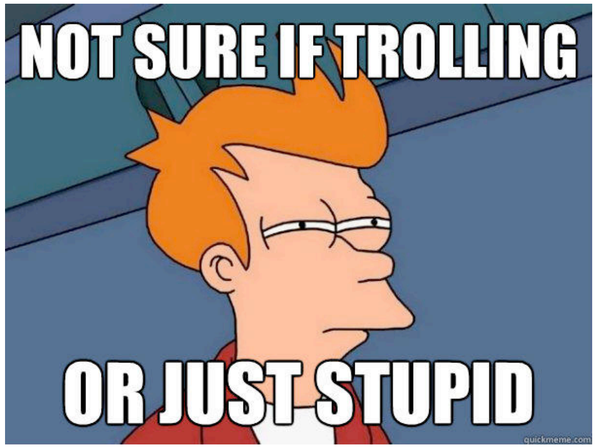
	Issue	Mitigating Best Practice
Typical	Weights are a little different	Some variation expected if training in parallel and on variety of hardware. Check within tolerance.
Typical	Weights are so different that model predictions are very different.	Training is getting stuck in local minima. A variety of training schema and hyper parameters and optimizers should be tried.
Tricky		
Tricky		
Tricky		

“The Reproducibility Problem” (in AI / ML)

- “I am able to train a model once, but I / someone else can’t reproduce the same model weights again.”

	Issue	Mitigating Best Practice
Typical	Weights are a little different	Some variation expected if training in parallel and on variety of hardware. Check within tolerance.
Typical	Weights are so different that model predictions are very different.	Training is getting stuck in local minima. A variety of training schema and hyper parameters and optimizers should be tried.
Tricky	Human mishandling, hard to detect	As model’s code is version controlled, **also version control model’s performance** so that performance results don’t get mixed up.
Tricky	Different datasets give different weights	This is to be expected within some tolerance. Just as model code is version controlled, **train/val/test datasets should be version controlled**.
Tricky	Works for me, but not for you	**Environment needs to be version controlled!** (Packages and versions)

How do you serve an AI controller model?



- We could just throw it on the machine and hope for the best!
- Scary reasons not to do that:
  - Data drift (incoming data is different from what the model was trained to do)
    - [other side: model performs poorly]
  - Stuff stops working and the accelerator operators throw away your “solution”
- MLOps can help!
  - Ok, so how do I know if this bad stuff happens? Data & performance monitoring! Alarming!
  - What do I do when this happens? Trigger workflows! Automate retraining! Deploy updated model!

Machine Learning Operations (MLOps)

- Deploying an AI/ML capability for operations requires more than data science (i.e. data discovery, labeling, and AI/ML model building).
- Deploying an AI/ML capability requires further engineering & stewardship:
  - Live-streaming / live-batched-streaming data ingestion and transformation
  - Model inference serving
  - Prediction streaming
    - Logging
    - Monitoring / triggering alarms
    - Automating actions

Workflow for MLOps

Data Management

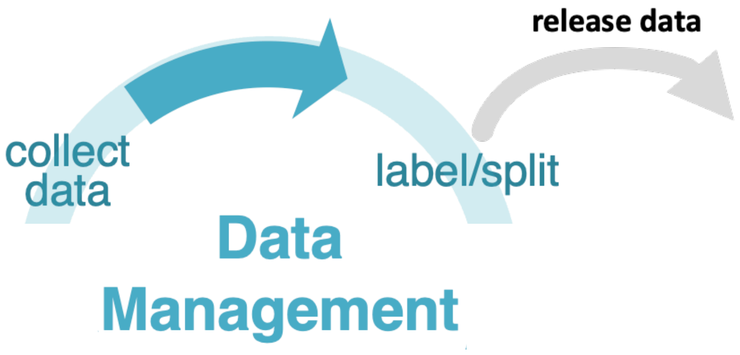
AI / ML Modeling

Operations Development

System Operations

Fermilab

Workflow for MLOps

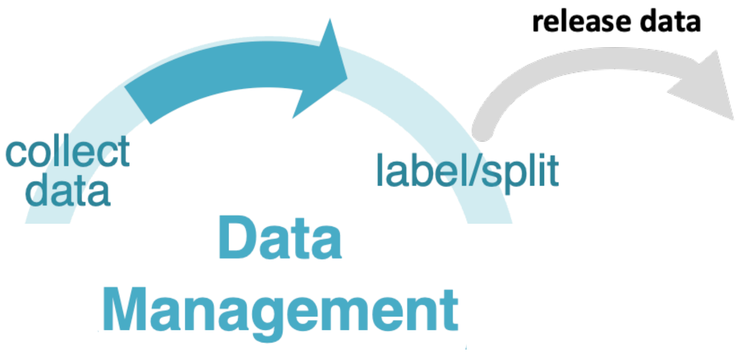


AI / ML Modeling

Operations Development

System Operations

Workflow for MLOps



AI / ML Modeling

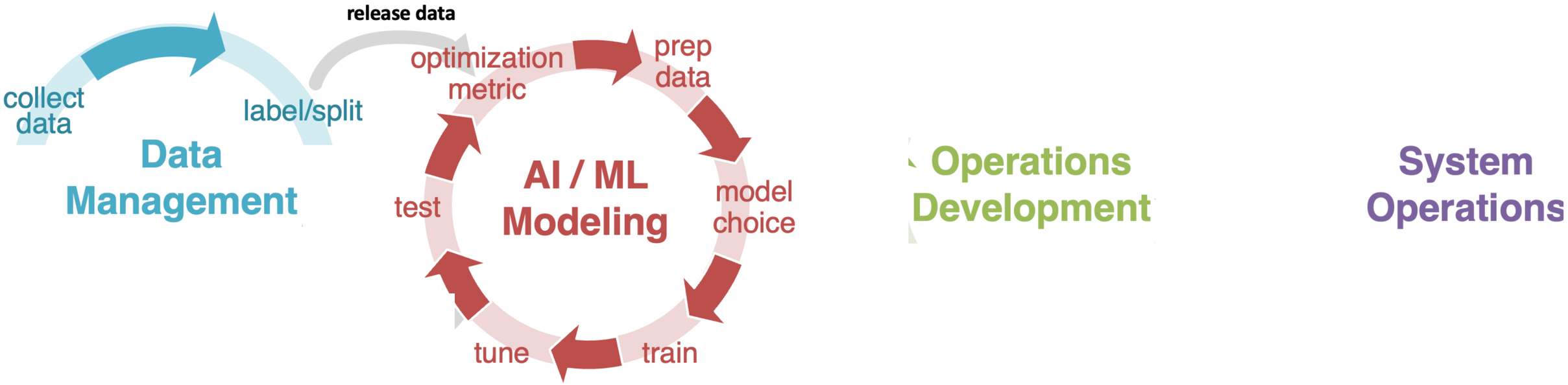
Operations Development

System Operations

- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- <u>Dataset Management System</u>
  - Versioning
  - Track derivative datasets
  - Metadata

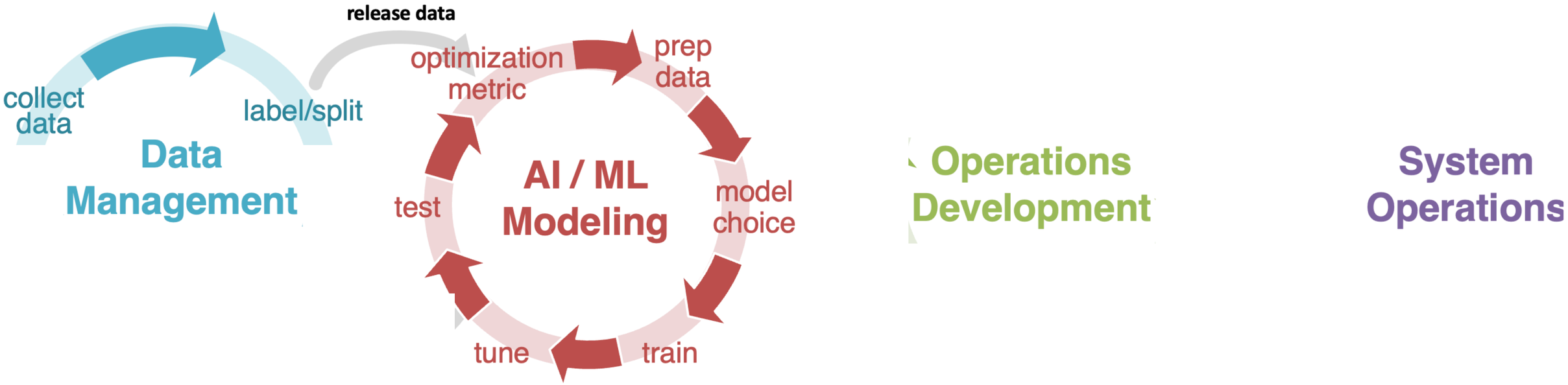
Fermilab

Workflow for MLOps



- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- <u>Dataset Management System</u>
  - Versioning
  - Track derivative datasets
  - Metadata

Workflow for MLOps



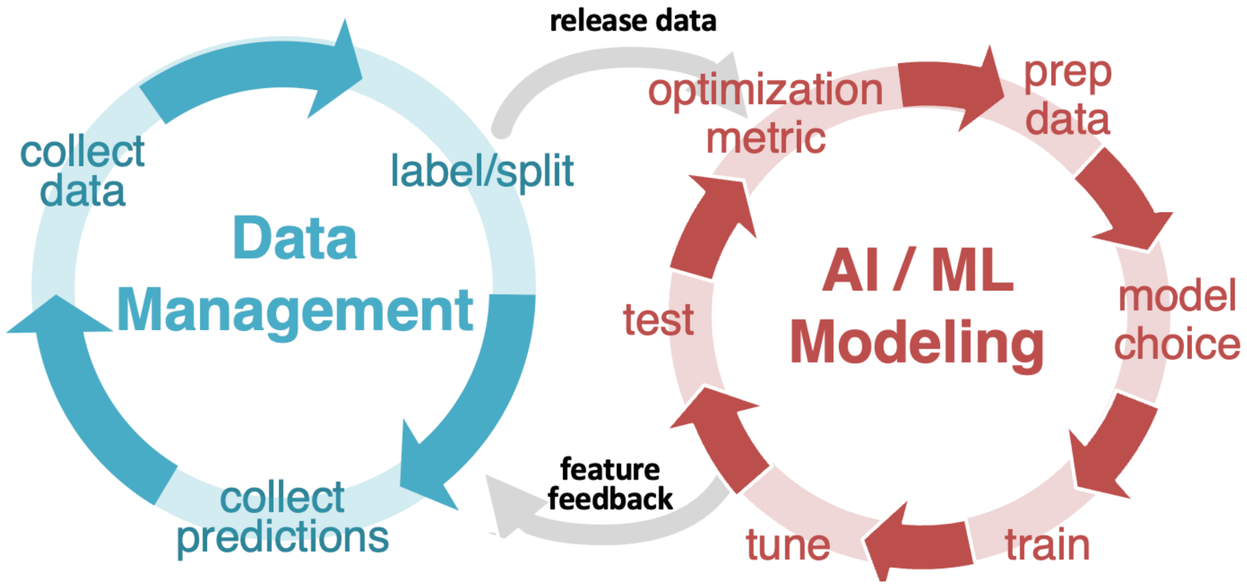
- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- <u>Dataset Management System</u>
  - Versioning
  - Track derivative datasets
  - Metadata

- Use Common Tools
- <u>Model Development System</u>
  - ~MLFlow / hyper p. tune
  - VC: model with references to env., data, results

-
  -
  -
  -
-
  -

-
-
-
-
-

Workflow for MLOps



Operations Development

System Operations

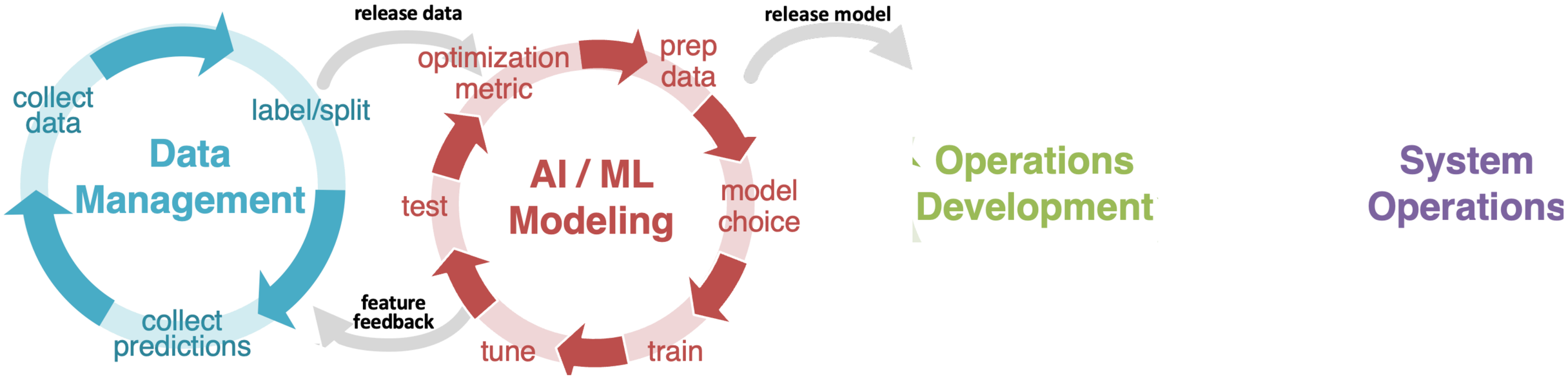
- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- Dataset Management System
  - Versioning
  - Track derivative datasets
  - Metadata

- Use Common Tools
- Model Development System
  - ~MLFlow / hyper p. tune
  - VC: model with references to env., data, results

-
  -
  -
  -
-
  -

-
-
-
-
-

Workflow for MLOps



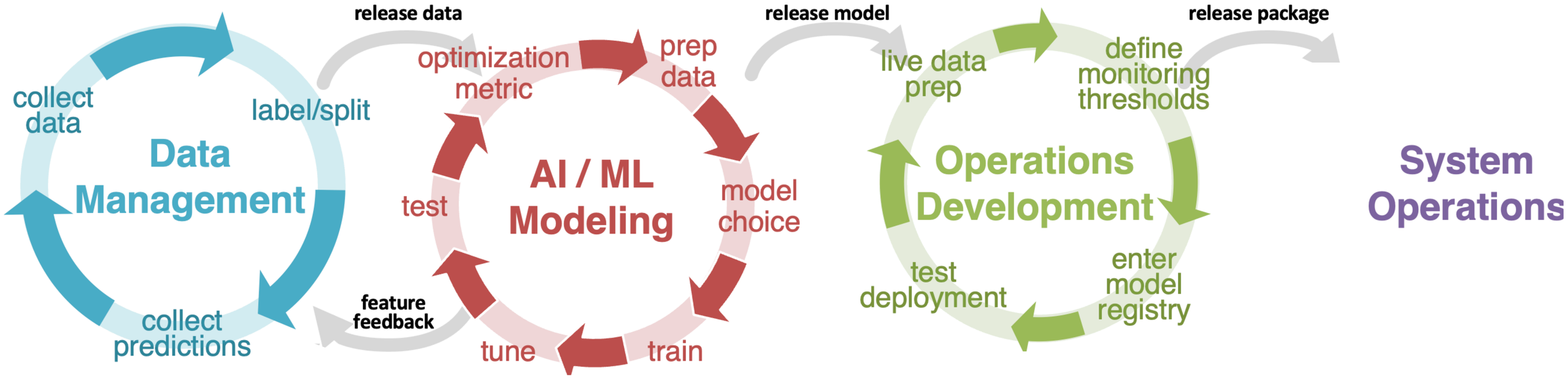
- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- <u>Dataset Management System</u>
  - Versioning
  - Track derivative datasets
  - Metadata

- Use Common Tools
- <u>Model Development System</u>
  - ~MLFlow / hyper p. tune
  - VC: model with references to env., data, results

-
  -
  -
  -
-
  -

-
-
-
-

Workflow for MLOps



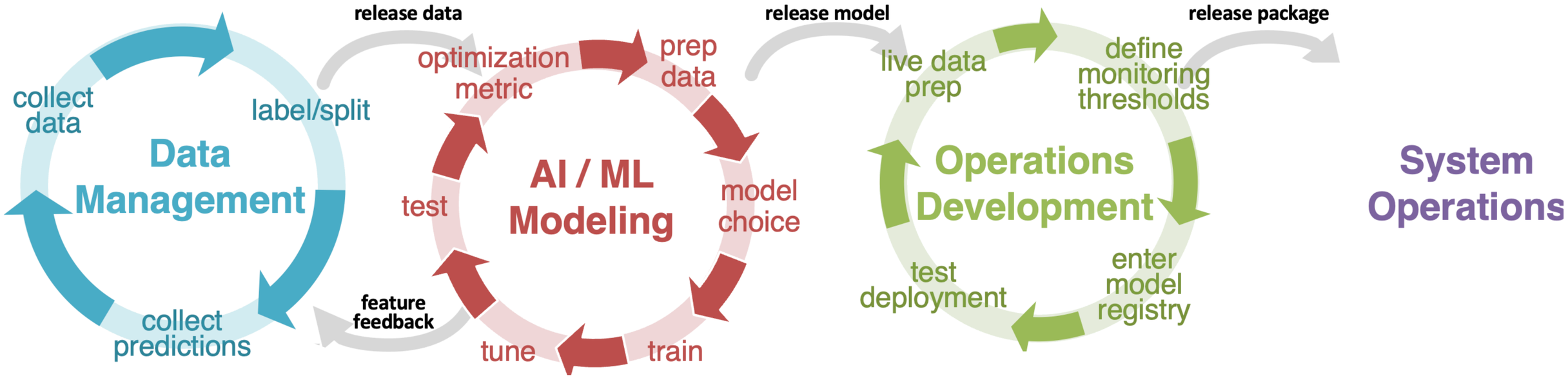
- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- Dataset Management System
  - Versioning
  - Track derivative datasets
  - Metadata

- Use Common Tools
- Model Development System
  - ~MLFlow / hyper p. tune
  - VC: model with references to env., data, results

-
  -
  -
  -
-
  -

-
-
-
-

Workflow for MLOps



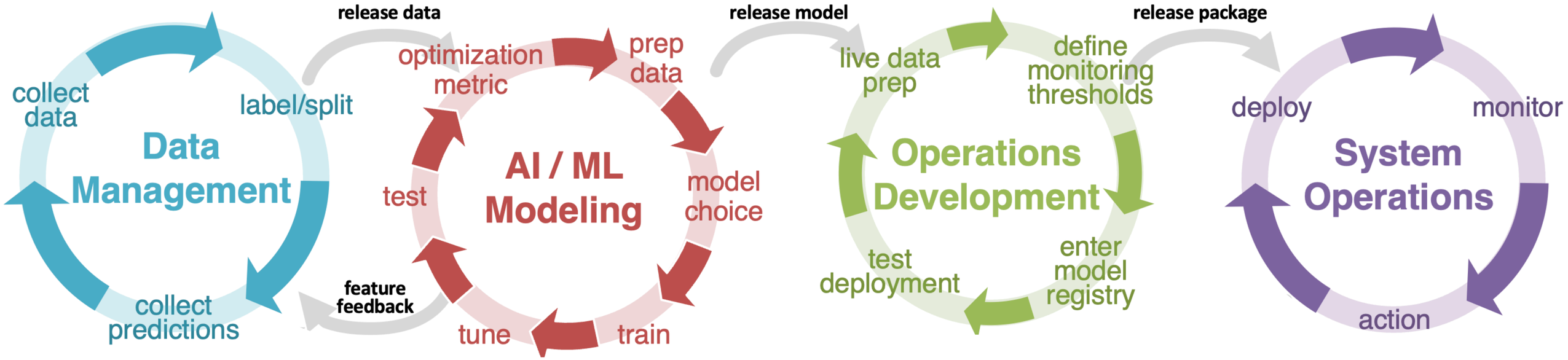
- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- Dataset Management System
  - Versioning
  - Track derivative datasets
  - Metadata

- Use Common Tools
- Model Development System
  - ~MLFlow / hyper p. tune
  - VC: model with references to env., data, results

- Interfaces for ML Engineer:
  - Getting “live data“
  - Setting “actions”
  - Monitoring input, model predictions/performance
- Model Registry
  - All data, env., model, and performance assets

-
-
-
-
-

Workflow for MLOps



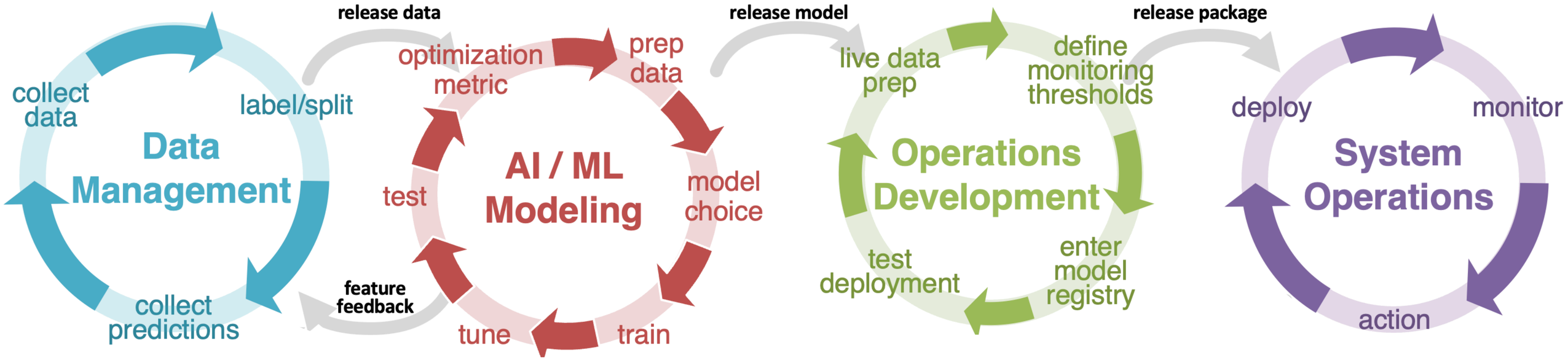
- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- <u>Dataset Management System</u>
  - Versioning
  - Track derivative datasets
  - Metadata

- Use Common Tools
- <u>Model Development System</u>
  - ~MLFlow / hyper p. tune
  - VC: model with references to env., data, results

- Interfaces for ML Engineer:
  - Getting “live data“
  - Setting “actions”
  - Monitoring input, model predictions/performance
- <u>Model Registry</u>
  - All data, env., model, and performance assets

-
-
-
-
-

Workflow for MLOps



- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- Dataset Management System
  - Versioning
  - Track derivative datasets
  - Metadata

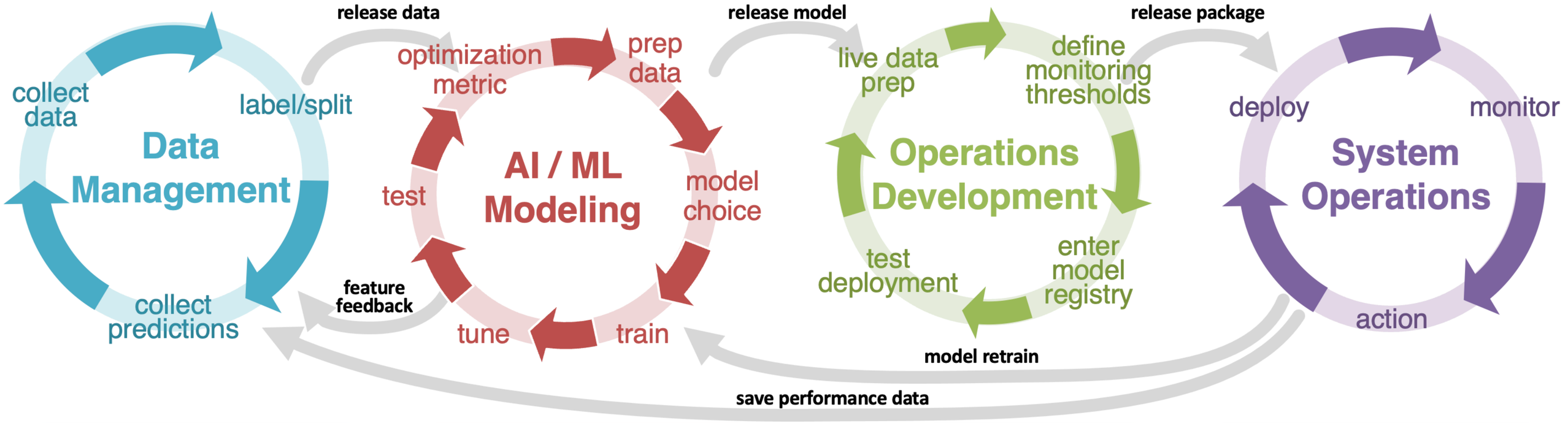
- Use Common Tools
- Model Development System
  - ~MLFlow / hyper p. tune
  - VC: model with references to env., data, results

- Interfaces for ML Engineer:
  - Getting “live data“
  - Setting “actions”
  - Monitoring input, model predictions/performance
- Model Registry
  - All data, env., model, and performance assets

- Server with proper specifications
- Monitoring services
- Control services
- (Logging services)
- Automate deployment of model assets from Model Registry

Workflow for MLOps

Automate as needed!



- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- Dataset Management System
  - Versioning
  - Track derivative datasets
  - Metadata

- Use Common Tools
- Model Development System
  - ~MLFlow / hyper p. tune
  - VC: model with references to env., data, results

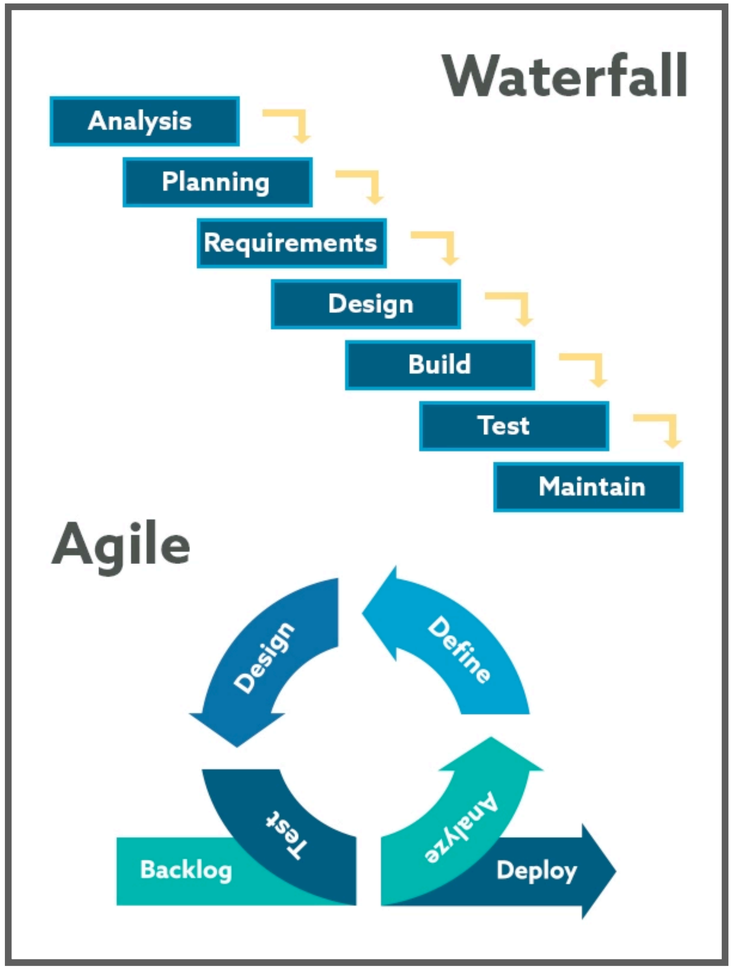
- Interfaces for ML Engineer:
  - Getting “live data“
  - Setting “actions”
  - Monitoring input, model predictions/performance
- Model Registry
  - All data, env., model, and performance assets

- Server with proper specifications
- Monitoring services
- Control services
- (Logging services)
- Automate deployment of model assets from Model Registry

MLOps is an expansion of DevOps

- MLOps = Machine Learning Operations
  - Play on DevOps: Development Operations
    - Integrate and streamline the development of software and its deployment
    - “Agile” software development practices
    - Continuous Integration / Continuous Delivery (CI/CD)
      - Modern code version control
      - Enforce strict permissions on merging
      - Enforce appropriate sized end-to-end tests
- Fun Fact: DevOps has roots from Lean Manufacturing practices ;)

Software Product Development Models



Infrastructure required for accelerator controls MLOps

Data Management

- Standardized accel. data logger & filter.
- Standardized format.
- Interface for ML Engineer: data filter.
- Dataset Management System
  - Versioning
  - Track derivative datasets
  - Metadata

Model Development

- Use Common Tools
- Model Development System
  - ~MLFlow / hyper p. tune
  - VC: model with references to env., data, results

Operations Dev.

- Interfaces for ML Engineer:
  - Getting “live data“
  - Setting “actions”
  - Monitoring input, model predictions/performance
- Model Registry
  - All data, env., model, and performance assets

System Operations

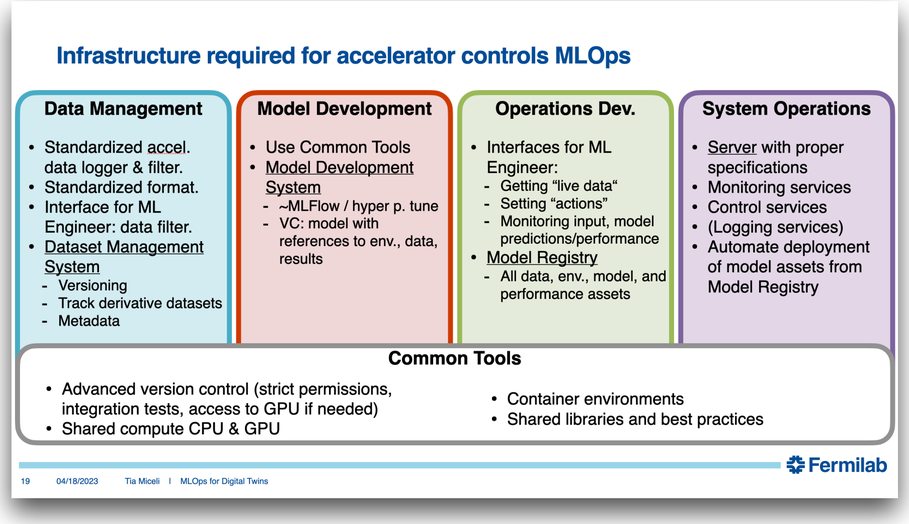
- Server with proper specifications
- Monitoring services
- Control services
- (Logging services)
- Automate deployment of model assets from Model Registry

Common Tools

- Advanced version control (strict permissions, integration tests, access to GPU if needed)
- Shared compute CPU & GPU
- Container environments, “code as infrastructure”
- Shared libraries and best practices

Fermilab is building out our accelerator controls MLOps

- Formalizing requirements for integration with current system & our modernized control system (ACORN 2020-2029)
- **Interviewing other accelerator laboratories**
  - Collab with SLAC on LUME-Services
  - Discussions with CERN
  - Discussions with BNL
  - Contacts at ORNL, ANL, please reach out!



- **Open source Toolset R&D on Kubernetes cluster**
  - Data management tools: Data lake, GraphQL, metadata database (PNNL DataHub, FNAL RUCIO, LinkedIn DataHub, Invenio)
  - Model development tools: MLFlow, DVC
  - Workflow tools: Airflow
  - Monitoring & control services: EPICS

★ Learned from presentation yesterday that we will consult with Tech Transfer Office about licensing!

Crowd-sourcing for the best solutions!

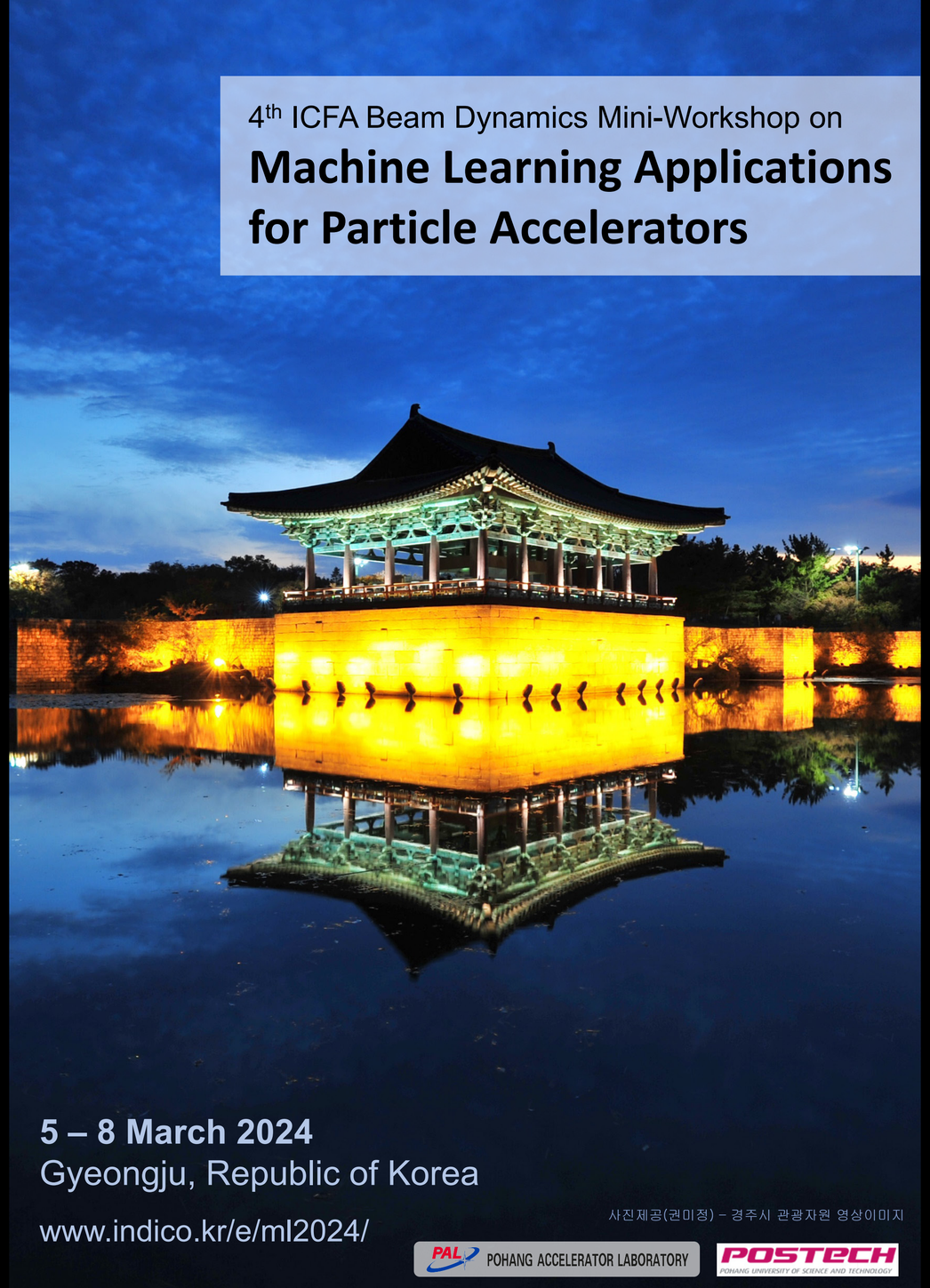
- Fermilab Accelerator Controls AI/ML Group is designing our MLOps infrastructure.
- I want to hear about your workflows
  - What worked?
  - What didn't?

Tia Miceli, Fermilab Accelerator AI/ML Group Lead a.k.a.
"Top Cat Herder in the Midwest!"

miceli@fnal.gov

4th ICFA Beam Dynamics Mini-Workshop on

Machine Learning Applications for Particle Accelerators



5 – 8 March 2024

Gyeongju, Republic of Korea

www.indico.kr/e/ml2024/

사진제공(권미정) – 경주시 관광자원 영상이미지

PAL POHANG ACCELERATOR LABORATORY

POSTECH POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY