Fermilab | U.S. DEPARTMENT OF ENERGY | Office of Science

Sustainable Implementation of Machine Learning for Particle Accelerators

Tia Miceli (Accelerator AI Coordinator)
Accelerator Reliability Workshop 2022
10/18/2022

Fermilab | U.S. DEPARTMENT OF ENERGY | Office of Science

Sustainable Implementation of Machine Learning for Particle Accelerators

Tia Miceli (Accelerator AI Coordinator)
Accelerator Reliability Workshop 2022
10/18/2022

… or **Best practices from Industry**;
where “Industry” := any business that uses ML in automating processes

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.
	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		

“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)

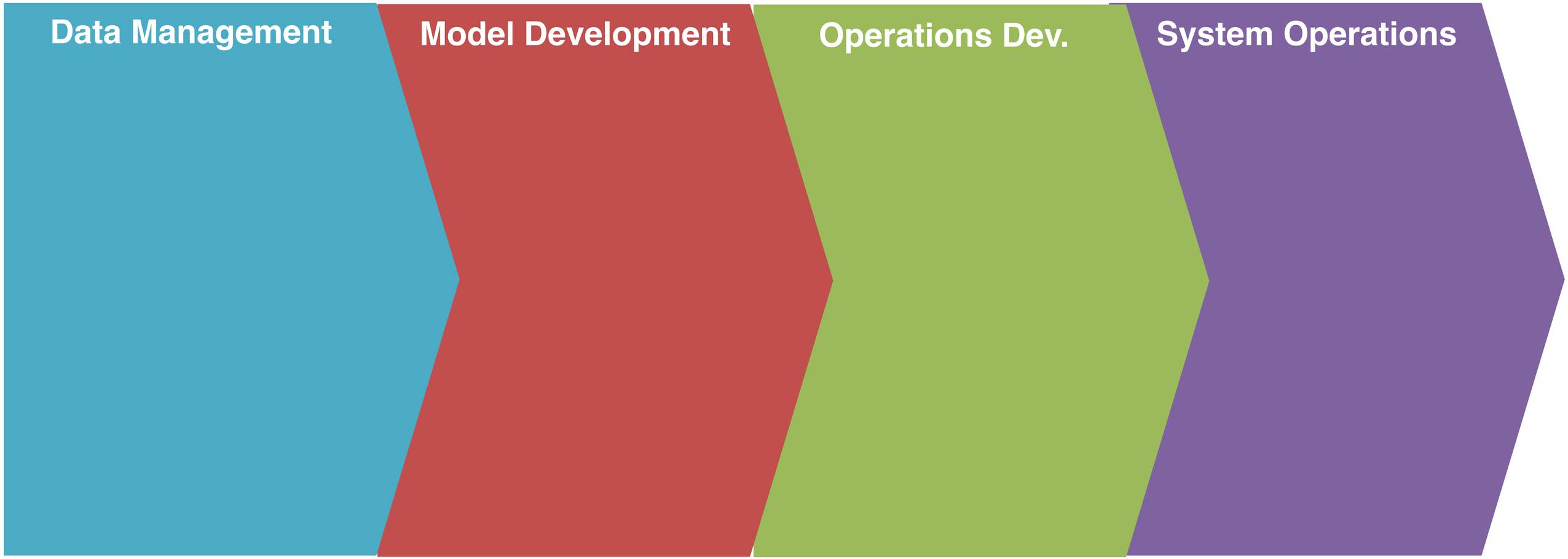
Machine Learning Operations (MLOps)

- 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

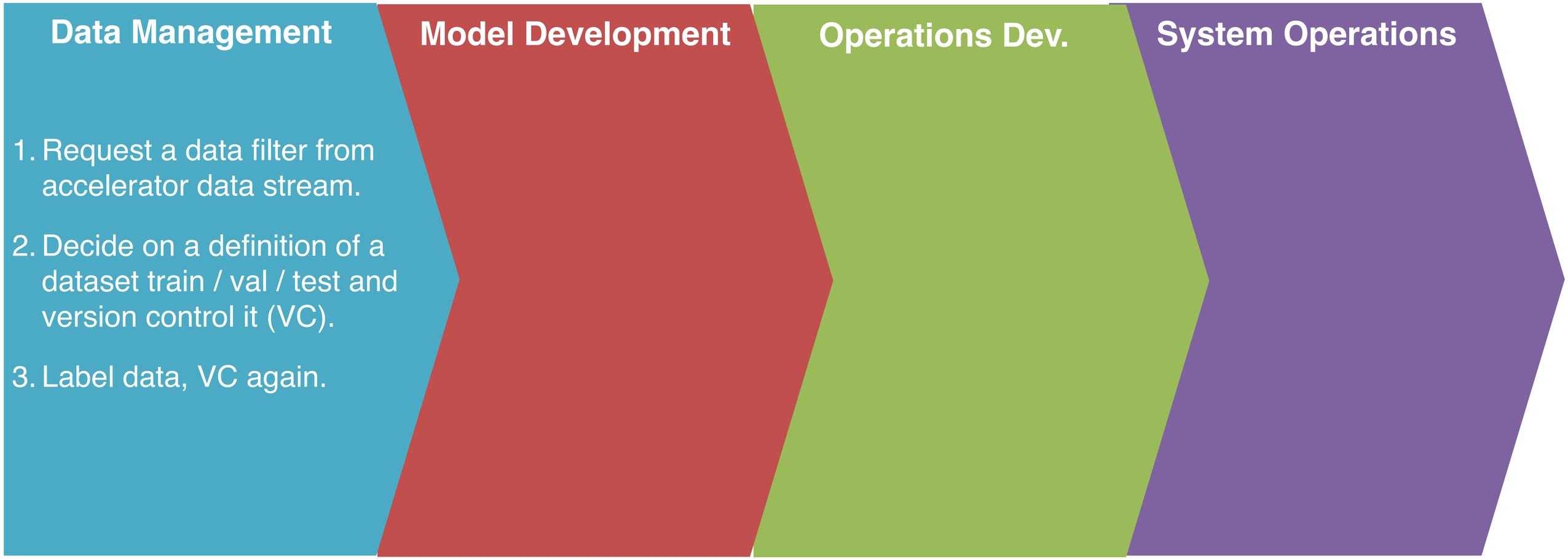
★ before describing the MLOps infrastructure, lets describe what user interactions with it should look like.

Model MLOps lifecycle from a ML Engineer's point of view

Assume we know that we have the I/O bandwidth and inference speed required to be effective.



Model MLOps lifecycle from a ML Engineer's point of view



Model MLOps lifecycle from a ML Engineer's point of view

Data Management

1. Request a data filter from accelerator data stream.
2. Decide on a definition of a dataset train / val / test and version control it (VC).
3. Label data, VC again.

Model Development

4. Choose metric(s) to optimize.
5. Prep / transform data of 2/3. VC code. (Optionally VC derivative dataset.)
6. Model trials (VC)
   - Model choice
   - Train
   - Tune
   - Test result (VC)
7. Save all model assets

Operations Dev.

System Operations

Model MLOps lifecycle from a ML Engineer's point of view

Data Management

1. Request a data filter from accelerator data stream.
2. Decide on a definition of a dataset train / val / test and version control it (VC).
3. Label data, VC again.

Model Development

4. Choose metric(s) to optimize.
5. Prep / transform data of 2/3. VC code. (Optionally VC derivative dataset.)
6. Model trials (VC)
   - Model choice
   - Train
   - Tune
   - Test result (VC)
7. Save all model assets

Operations Dev.

8. Prepare transforms for live data to model.
9. Choose thresholds for monitoring:
   - Live input data
   - Prediction / performance
10. Enter model registry
11. Test deployment

System Operations

Model MLOps lifecycle from a ML Engineer's point of view

Data Management

1. Request a data filter from accelerator data stream.
2. Decide on a definition of a dataset train / val / test and version control it (VC).
3. Label data, VC again.

Model Development

4. Choose metric(s) to optimize.
5. Prep / transform data of 2/3. VC code. (Optionally VC derivative dataset.)
6. Model trials (VC)
   - Model choice
   - Train
   - Tune
   - Test result (VC)
7. Save all model assets

Operations Dev.

8. Prepare transforms for live data to model.
9. Choose thresholds for monitoring:
   - Live input data
   - Prediction / performance
10. Enter model registry
11. Test deployment

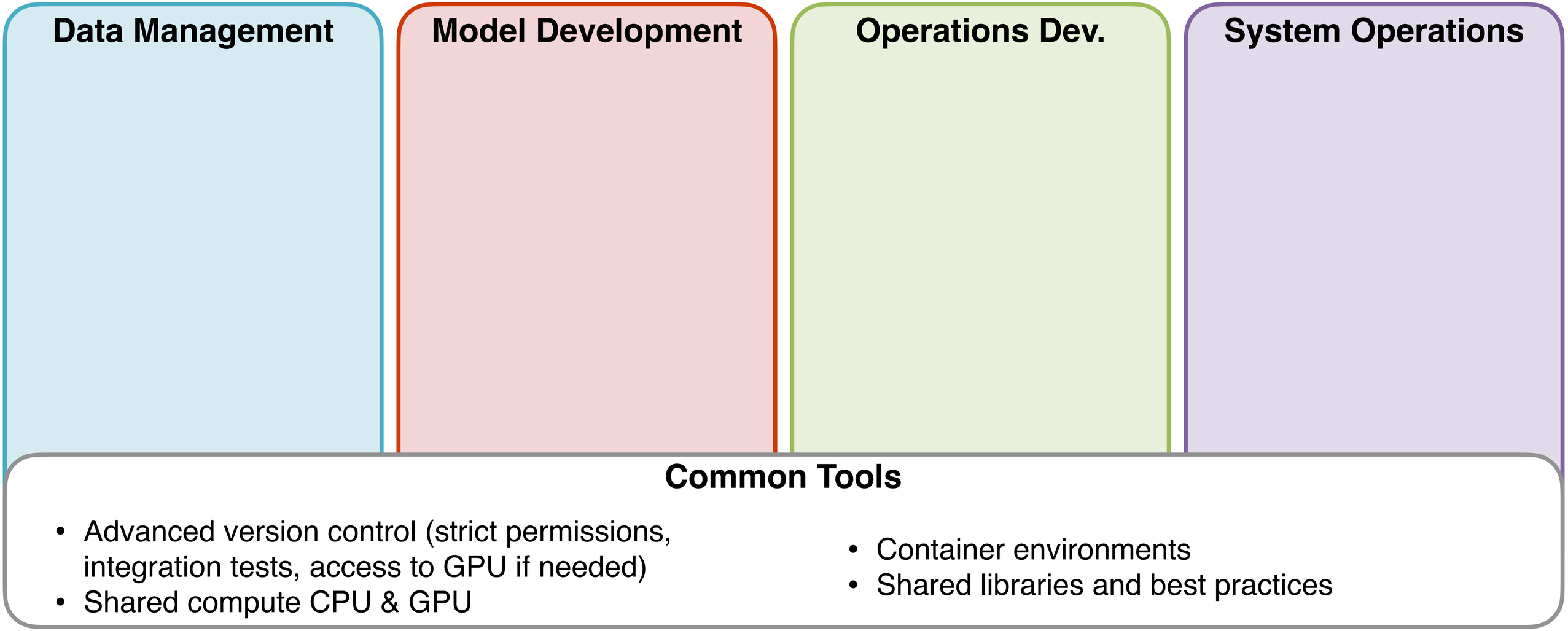
System Operations

12. Periodic model registry inventory
13. Respond to alarms:
    - **Adjust thresholds** (go to 9 onward), can be automated.
    - **Adjust model weights** (go to 1-3, 6,7, 9 onward), can be automated (RL).
    - **Retire & replace**

Key pieces of Continuous Integration / Continuous Delivery to include

- Use a modern code version control system
- Enforce strict permissions on merging
- Enforce unit tests of small functions before merging
- Enforce end-to-end (integration) tests before merging
- Optionally enforce coding style choices

Infrastructure required for accelerator controls MLOps



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

Operations Dev.

System Operations

Common Tools

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

Infrastructure required for accelerator controls MLOps

Data Management

- Standardized accel. data logger / filter.
- Standardized format.
- 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.

System Operations

Common Tools

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

Infrastructure required for accelerator controls MLOps

Data Management

- Standardized accel. data logger / filter.
- Standardized format.
- 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

Common Tools

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

Infrastructure required for accelerator controls MLOps

Data Management

- Standardized accel. data logger / filter.
- Standardized format.
- 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
- Shared libraries and best practices

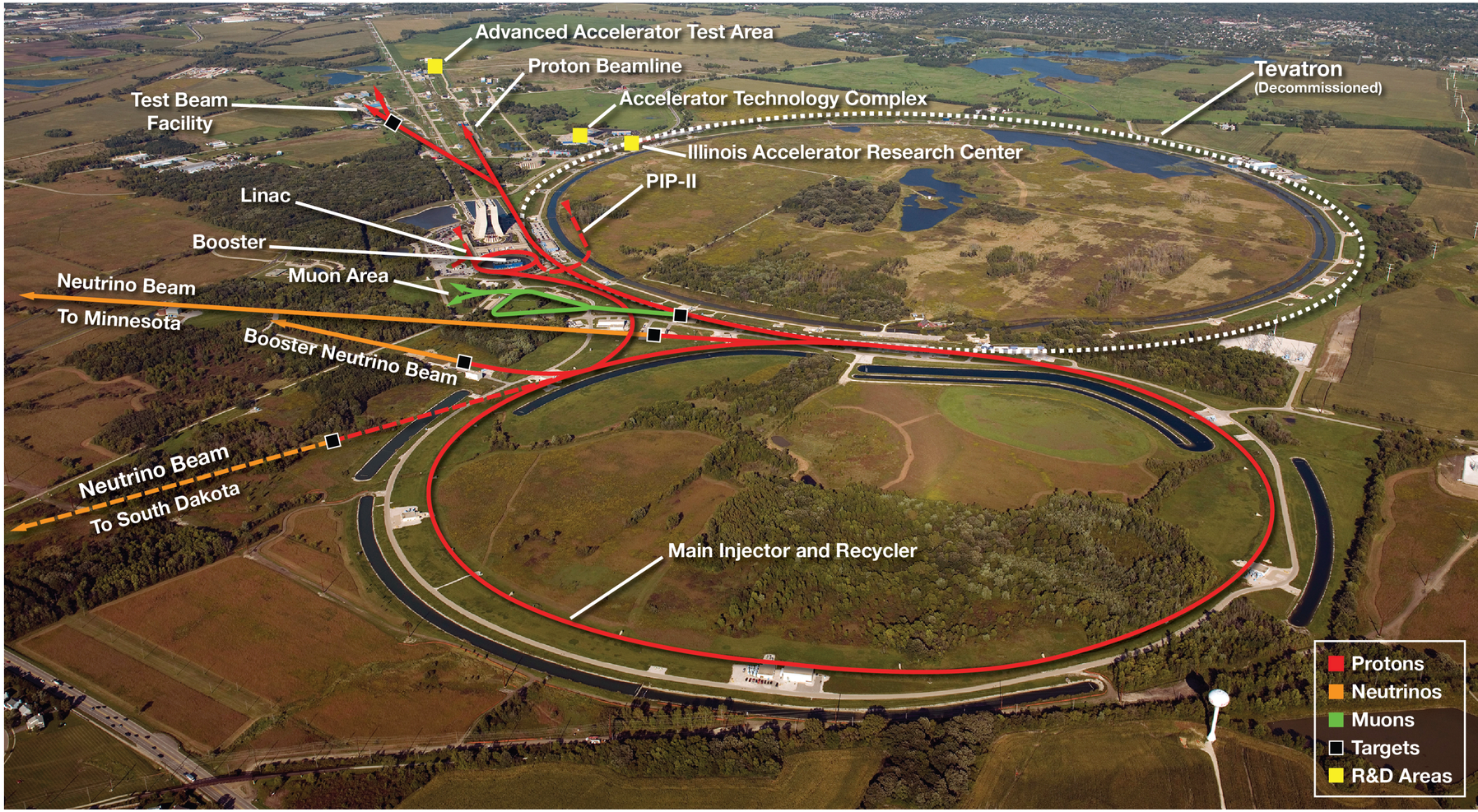
Fermilab is on this path!

- We've secured shared compute at our Elastic Analysis Facility
  - Jupyter Hub on Kubernetes with managed environments
  - ~225 CPU and 1.1 TB memory + 45 TB storage (for now)
- We've secured an on-premise GitHub install, soon to be GitHub Cloud
- We've secured a modern 😉 content management system (Confluence Cloud)
- These were the low hanging fruit, the rest requires more requirements gathering and R&D, so we are collaborating with Fermilab's ACORN project.

Accelerator Controls Operations Research Network (ACORN)

- ACORN is a DOE O413 project that will **modernize Fermilab's accelerator control system** by replacing obsolete hardware and software and will integrate the new control system with our new SRF linac* and neutrino beamline**.
- CD-0 Aug 28, 2020
- CD-1 projected Q2 2024

Part of **modernization** is to enable AI / ML for accelerator controls!



* Proton Improvement Plan II (PIP-II)

** Long Baseline Neutrino Facility beamline (LBNF)

Fermilab

ACORN R&D projects underway that will help enable AI / ML

- **Data Logger**: What data is chosen? (All please!) How long does the data last? How much data is logged?
- **Curated Data Storage**: How does data get stored? How does data get transformed to formats needed in different contexts?

Data Lake Concept: All data stored in native format and transformed on demand

⇒ These will help create a strong base for MLOps: Data Management

Conclusion

- MLOps best practices ensure accelerator controls AI / ML remains sustainable.
- With Fermilab's ACORN project we are currently gathering requirements and doing R&D to define our optimal MLOps infrastructure.
- I would love to learn about systems in place at other accelerators! Please come chat with me!
  - miceli@fnal.gov

Tia Miceli, Fermilab Accelerator Controls AI Coordinator
a.k.a. "Top Cat Herder in the Midwest!"

Special thanks to ARW2022 and to Fermilab's ACORN Project!