MATLAB EXPO

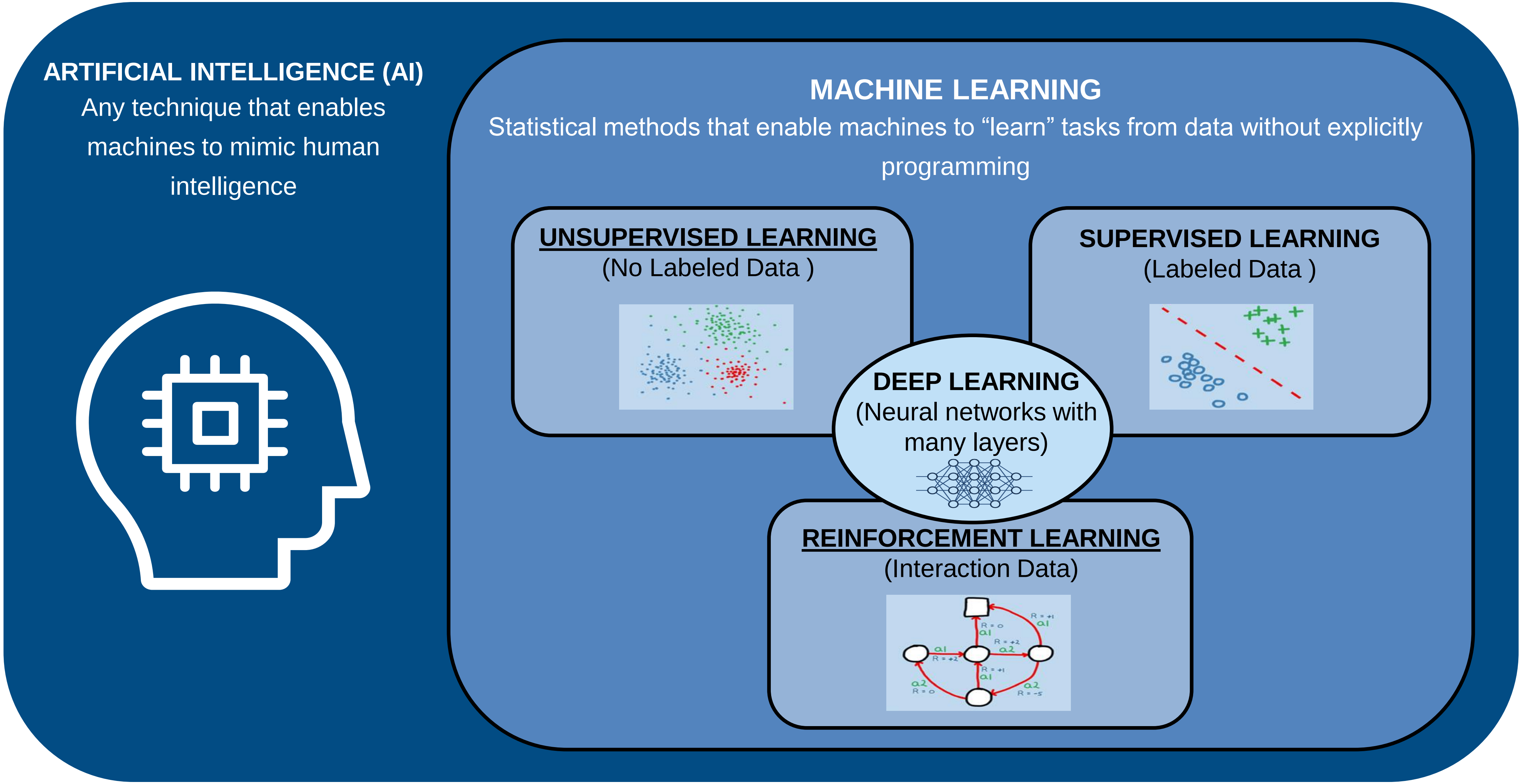
Scaling Artificial Intelligence: From Model Development to Operationalization

Dr Rishu Gupta, MathWorks

Peeyush Pankaj, MathWorks

MathWorks

Machine Learning is a key technology driving the AI megatrend



Brief Overview for AI-driven system design

Data Preparation	AI Modeling	Simulation & Test	Deployment
Data cleansing and preparation	Model design and tuning	Integration with complex systems	Embedded devices
Human insight	Hardware accelerated training	System simulation	Enterprise systems
Simulation-generated data	Interoperability	System verification and validation	Edge, cloud, desktop

Data Preparation
- **Labeller apps**
- **Unreal co-simulation**
- **Simscape**

AI Modeling
- **Deep network designer**
- **Experiment manager**
- **Classification learner**

Simulation & Test
- **Simulink as integration platform**

Deployment
- **CPUs, GPUs, FPGAs**
- **Cloud (on-premise, service providers)**

2

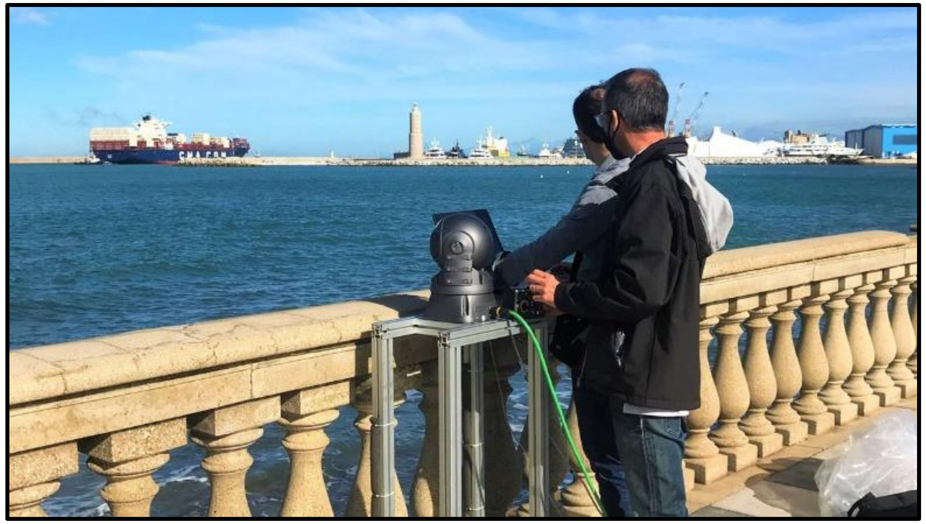
Drass Develops Deep Learning System for Real-Time Object Detection in Maritime Environments

Challenge

Help ship operators monitor sea environments and detect objects, obstacles, and other ships

Solution

Create an object-detection deep learning model that can be deployed on ships and run-in real time



First day of object detection tests with optronic system prototype.

Results

- Data labeling automated
- Development time reduced
- Flexible and reproducible framework established

"From data annotation to choosing, training, testing, and fine-tuning our deep learning model, MATLAB had all the tools we needed—and GPU Coder enabled us to rapidly deploy to our NVIDIA GPUs even though we had limited GPU experience."

- Valerio Imbriolo, Drass Group

[Link to user story]

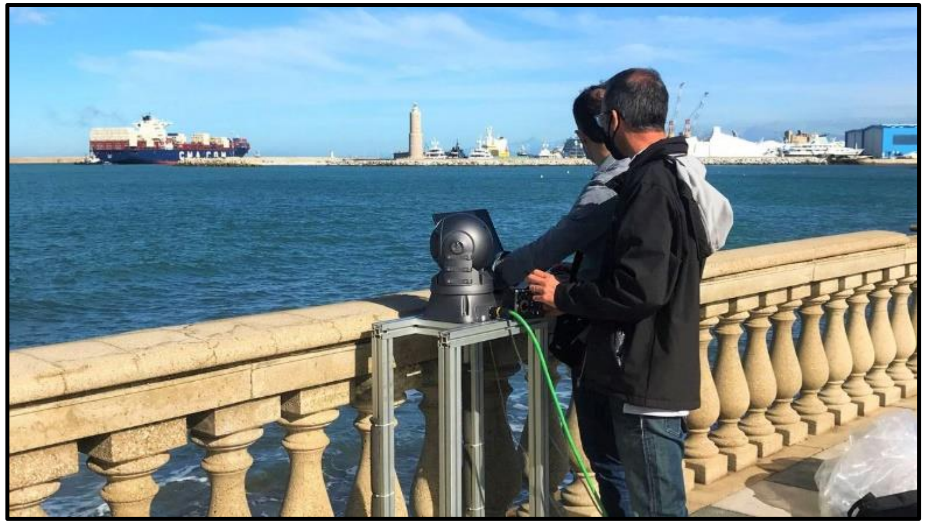
Drass Develops Deep Learning System for Real-Time Object Detection in Maritime Environments

Challenge

Help ship operators monitor sea environments and detect objects, obstacles, and other ships

Solution

Create an object-detection deep learning model that can be deployed on ships and run-in real time



First day of object detection tests with optronic system prototype.

Results

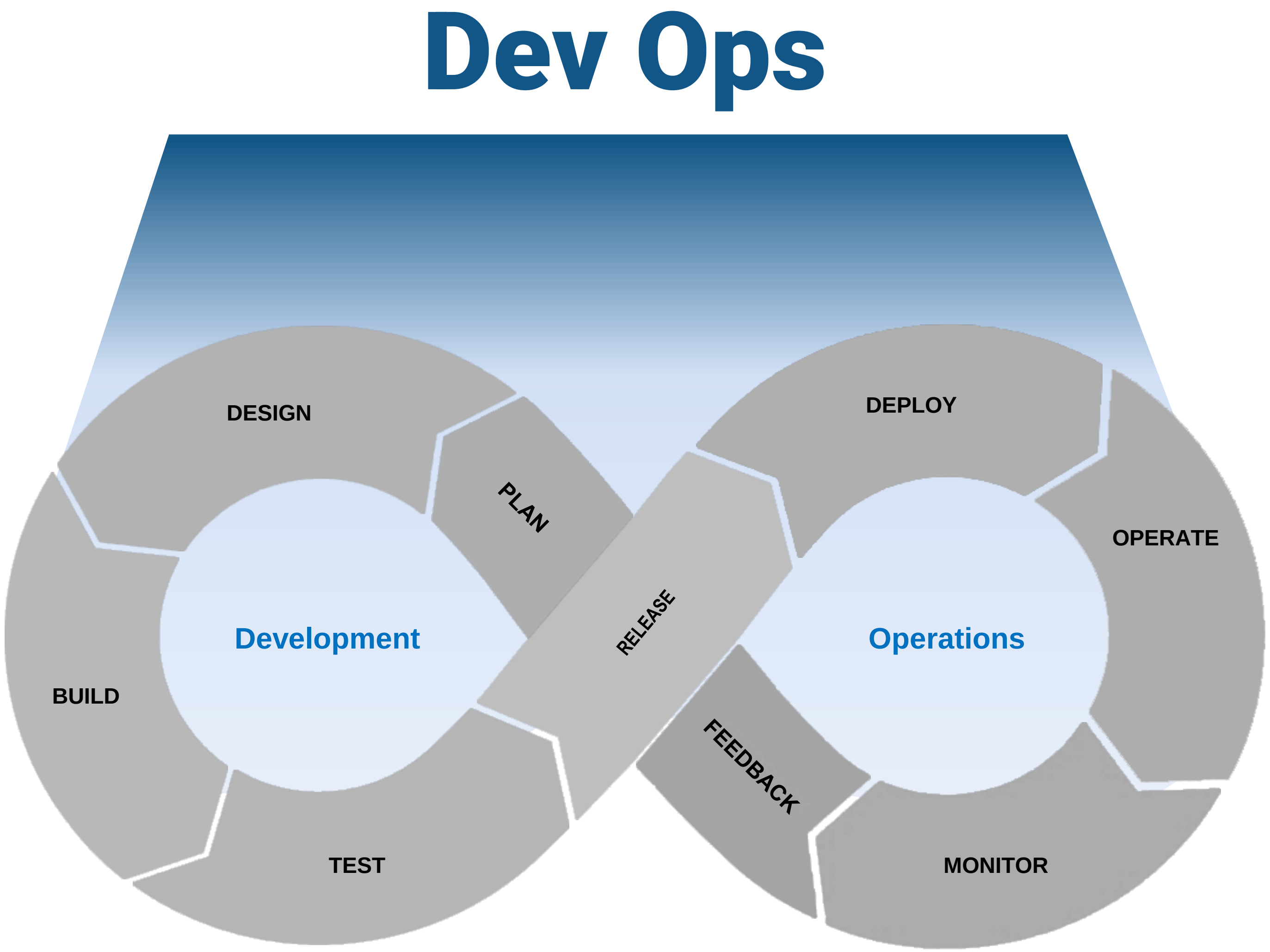
- Data labeling automated **(From 3 mins per frame to 0.3 secs per frame)**
- Development time reduced **(From 18 months to 10 months)**
- Flexible and reproducible framework established **(modify, retrain, update and reintegrate with minimal effort)**

"From data annotation to choosing, training, testing, and fine-tuning our deep learning model, MATLAB had all the tools we needed—and GPU Coder enabled us to rapidly deploy to our NVIDIA GPUs even though we had limited GPU experience."

- Valerio Imbriolo, Drass Group

Link to user story

Scaling AI-driven systems



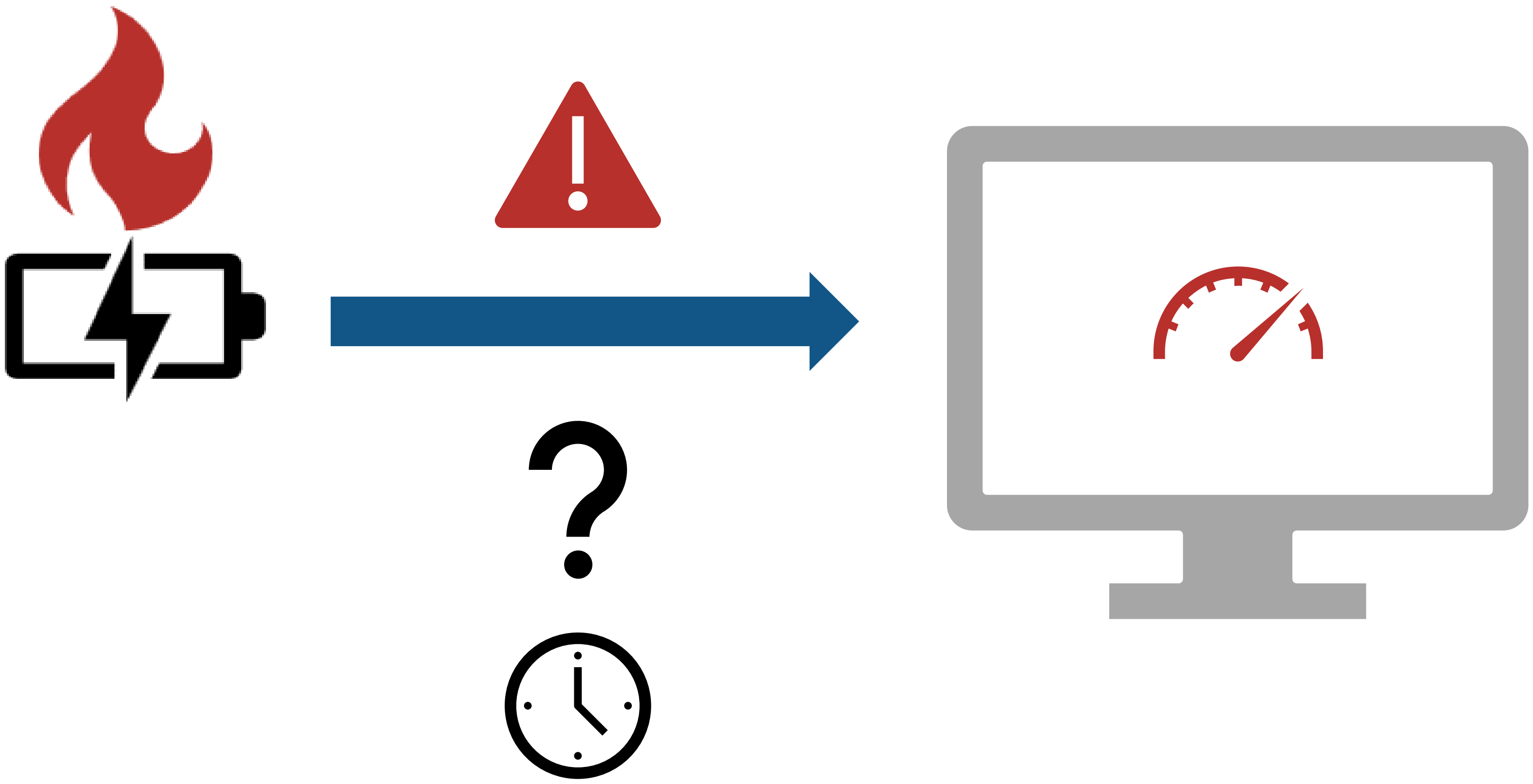
Electric batteries are everywhere. Effective management increases vehicle availability and reduces costs.

Hybrid electric city bus

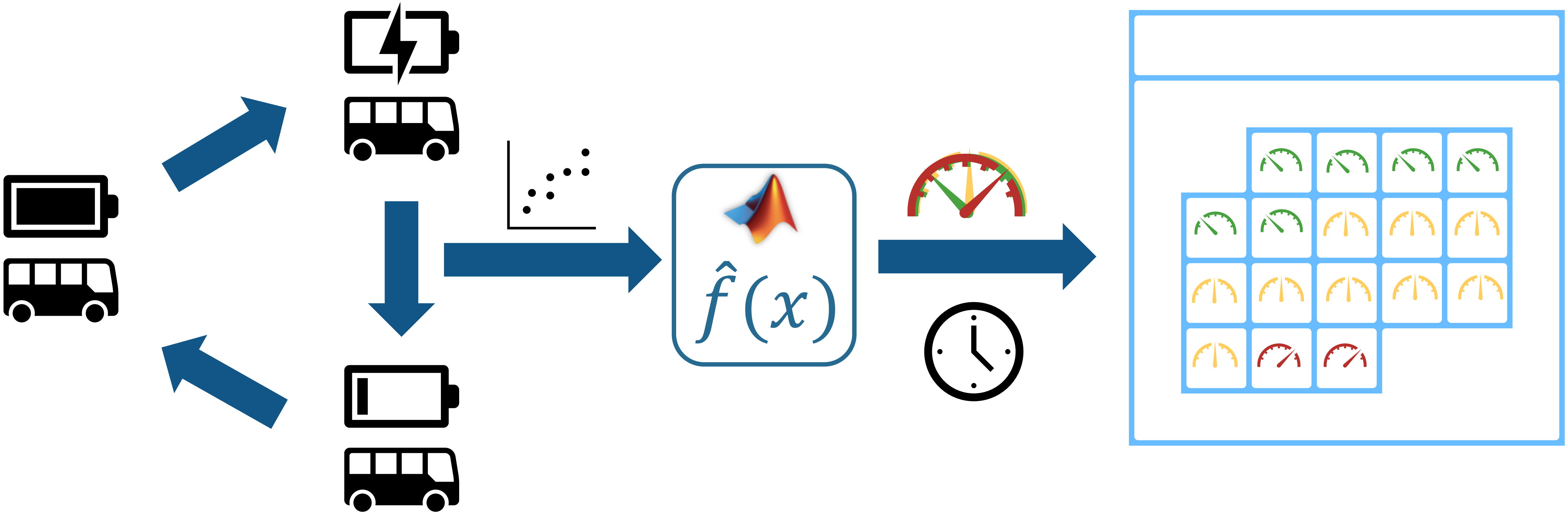
Autonomous electric tractor

Industrial robots

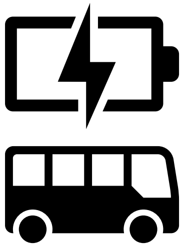
Monitoring battery health is good. Predicting it is better.

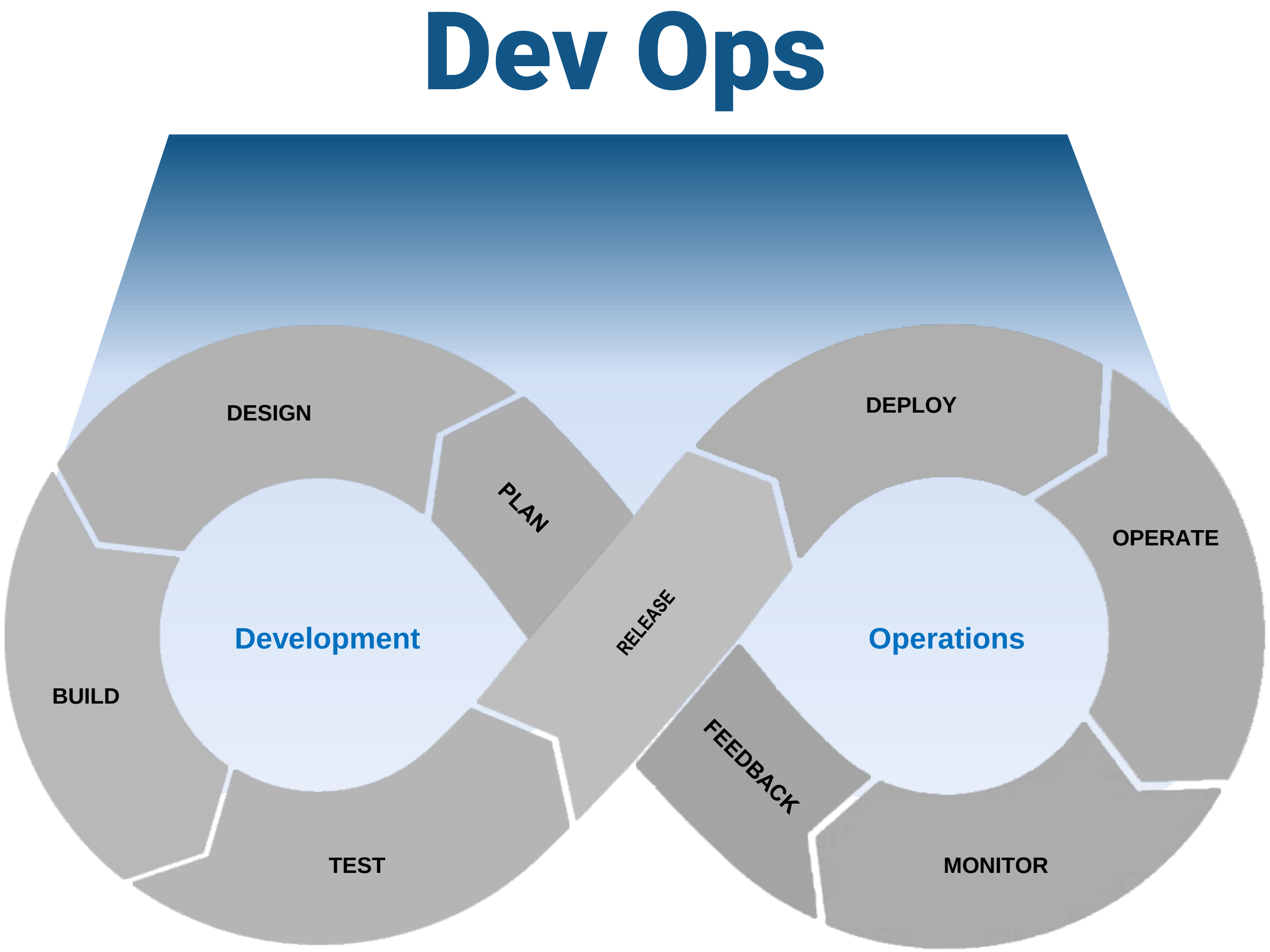


Predictive maintenance enables downtime to be scheduled rather than disruptive.



Understanding the lifecycle of a machine learning solution lets you know if you've automated all of it.

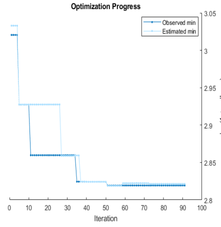




Automating development requires deep knowledge of the domain.

Design

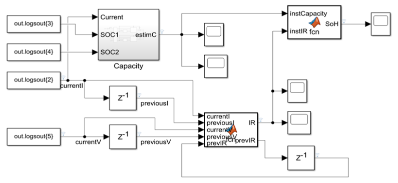
“AutoML”



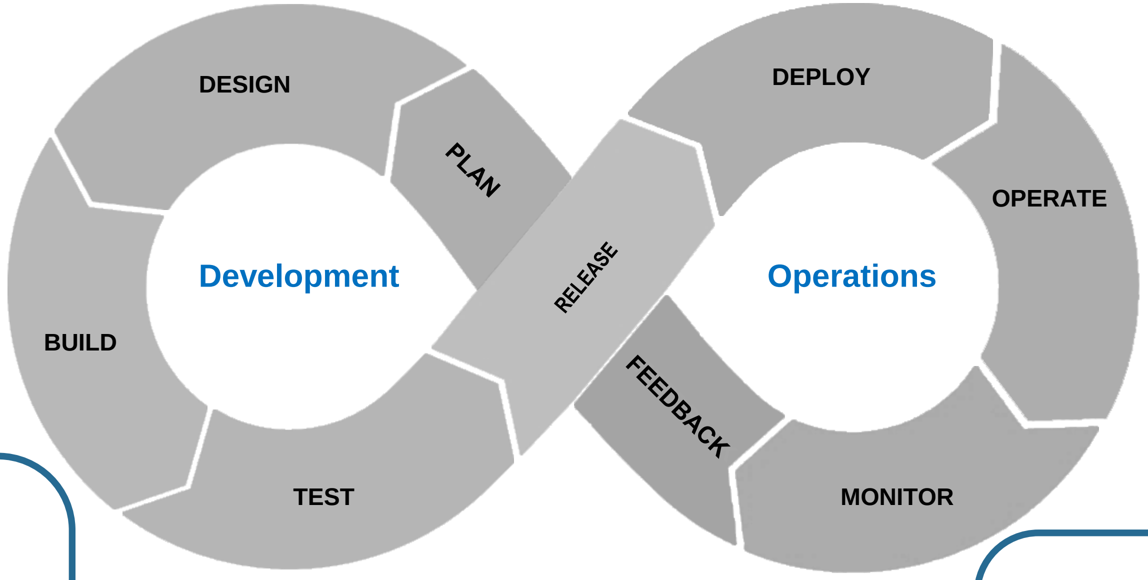
Build

Test

Model-based labeling



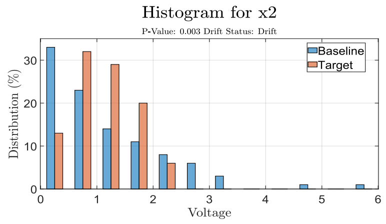
Deploy



$\hat{f}(x)$

Operate

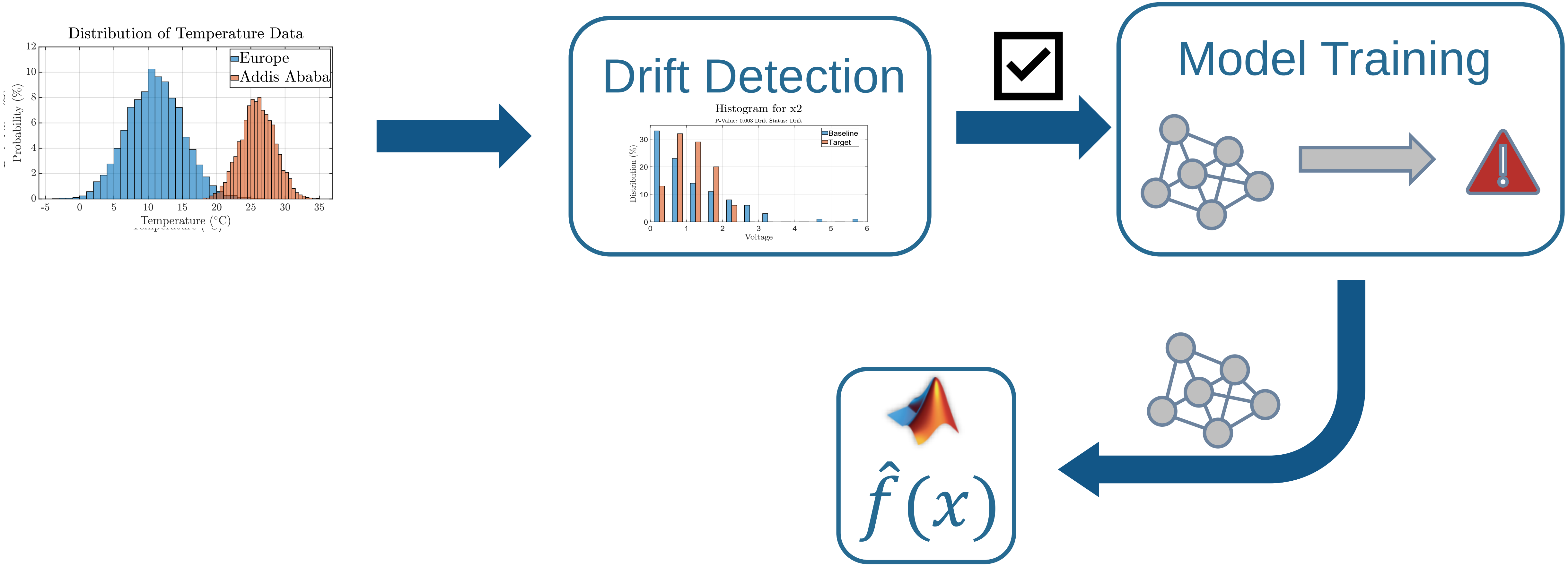
Drift Detection



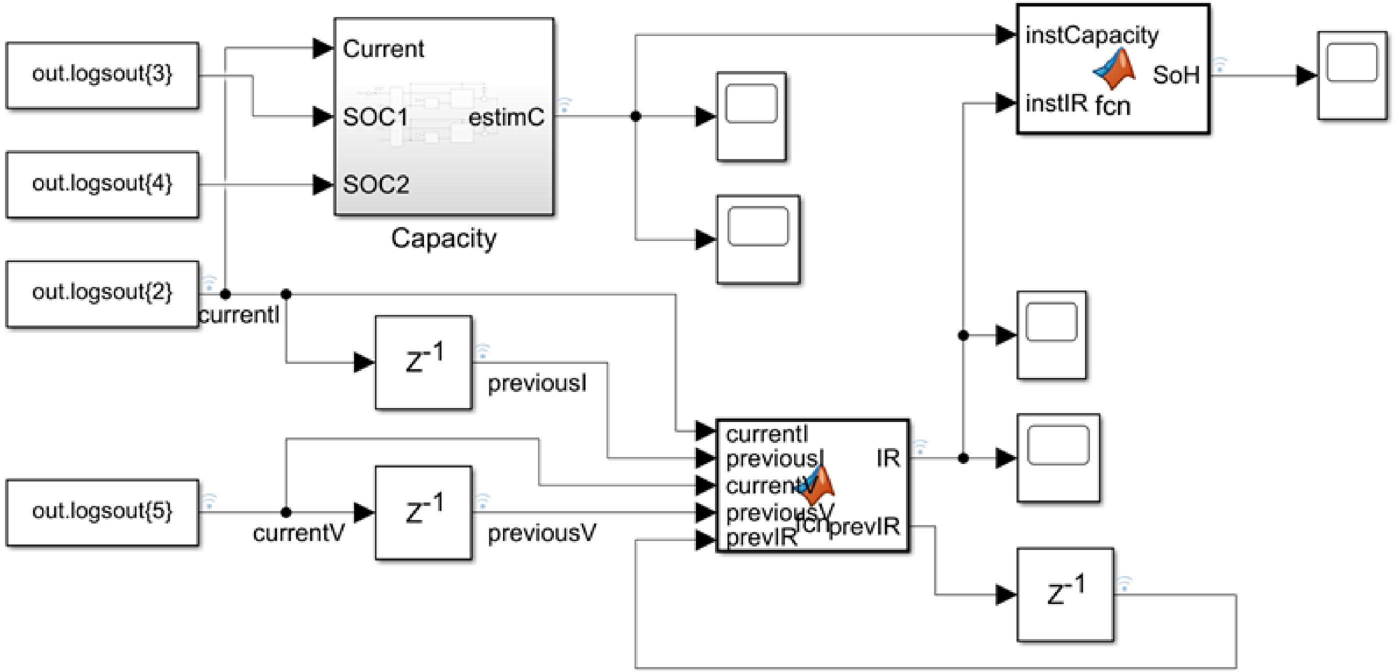
Monitor

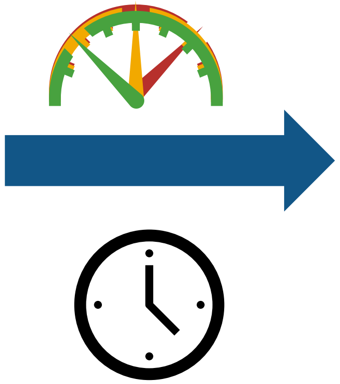
Feedback

Automatic drift detection compares the observed data to the training data to determine when retraining is required.

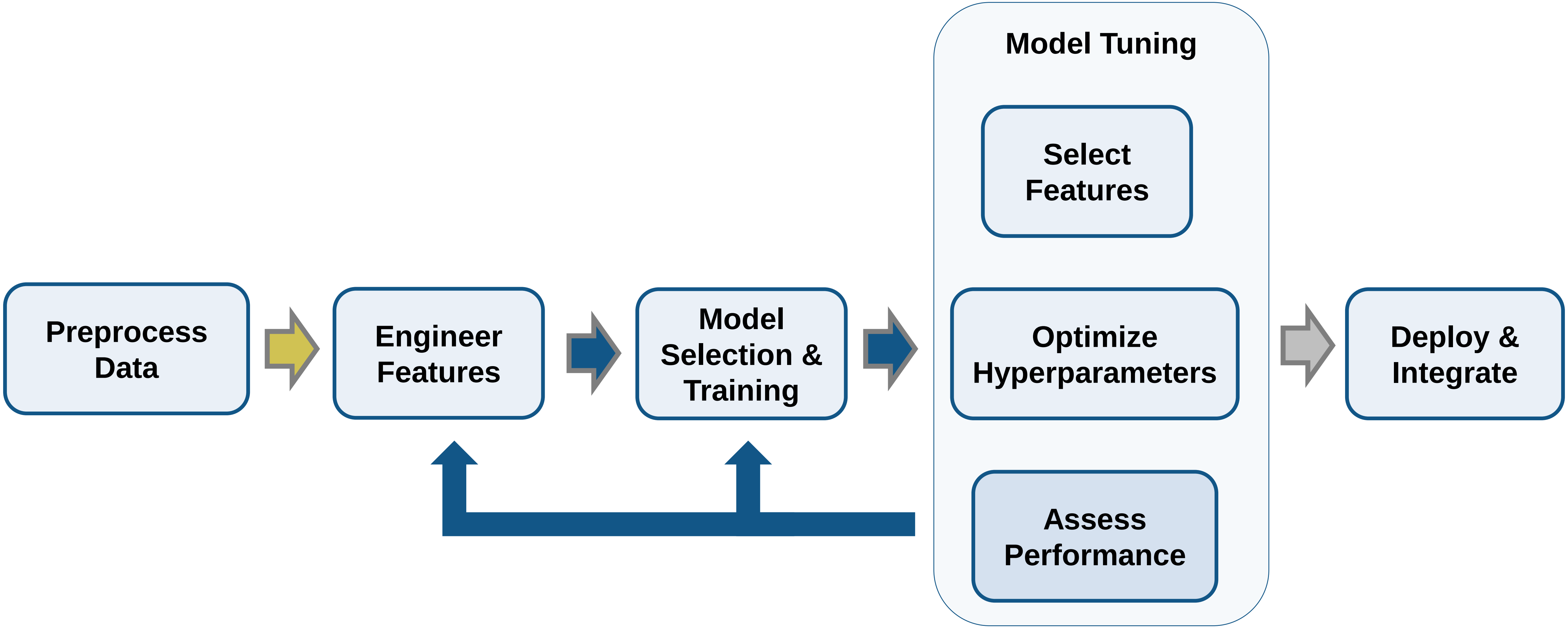


High-fidelity physical models accurately label observed data.



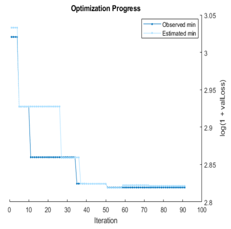


AutoML selects the model and hyperparameters that perform best on the drifting data.

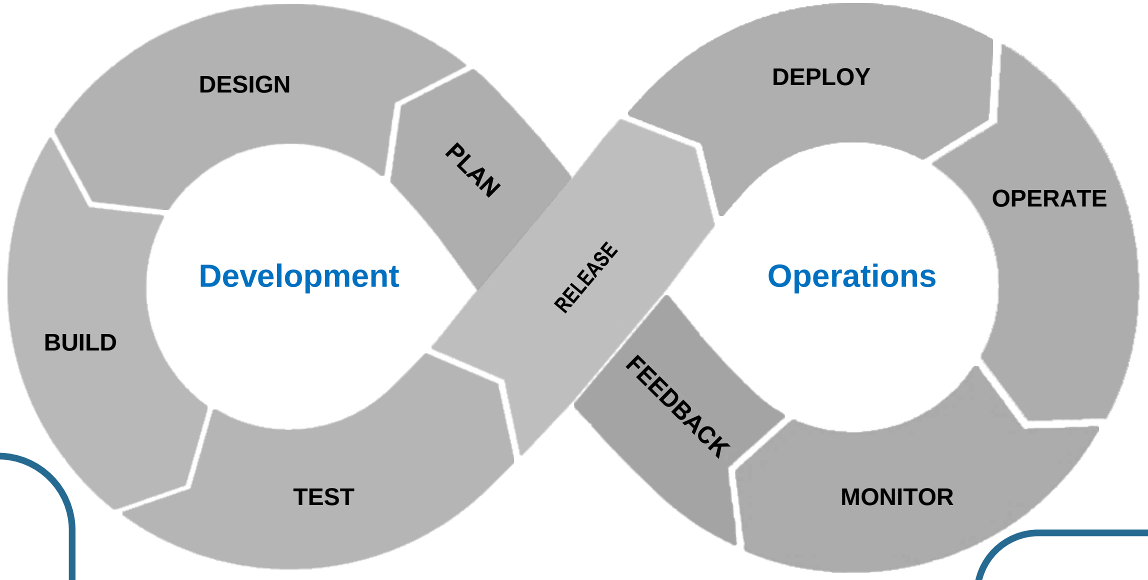


Let us remind ourselves the blueprint of the automated solution.

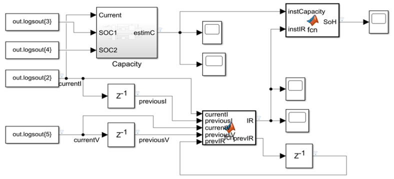
"AutoML"



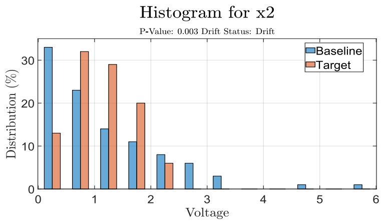
$$\hat{f}(x)$$



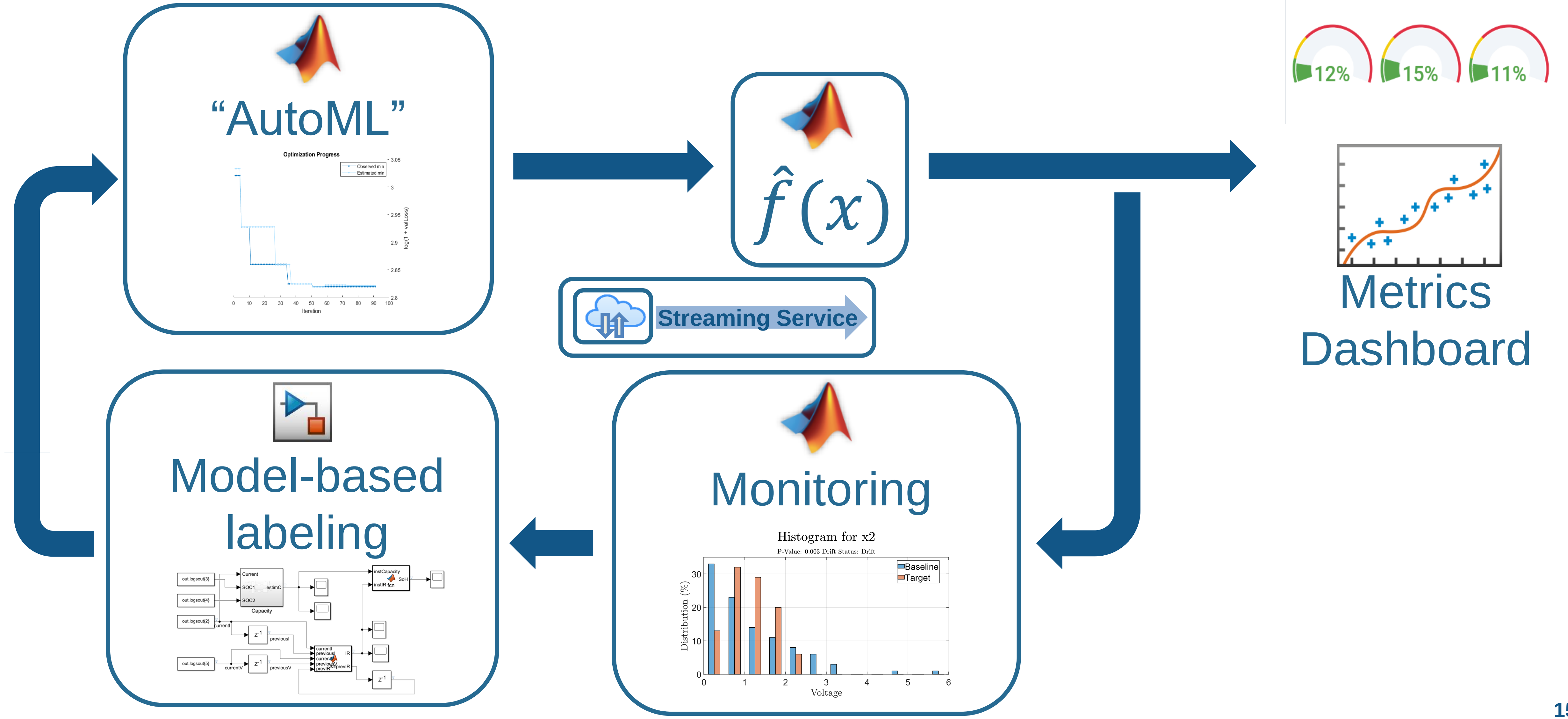
Model-based labeling



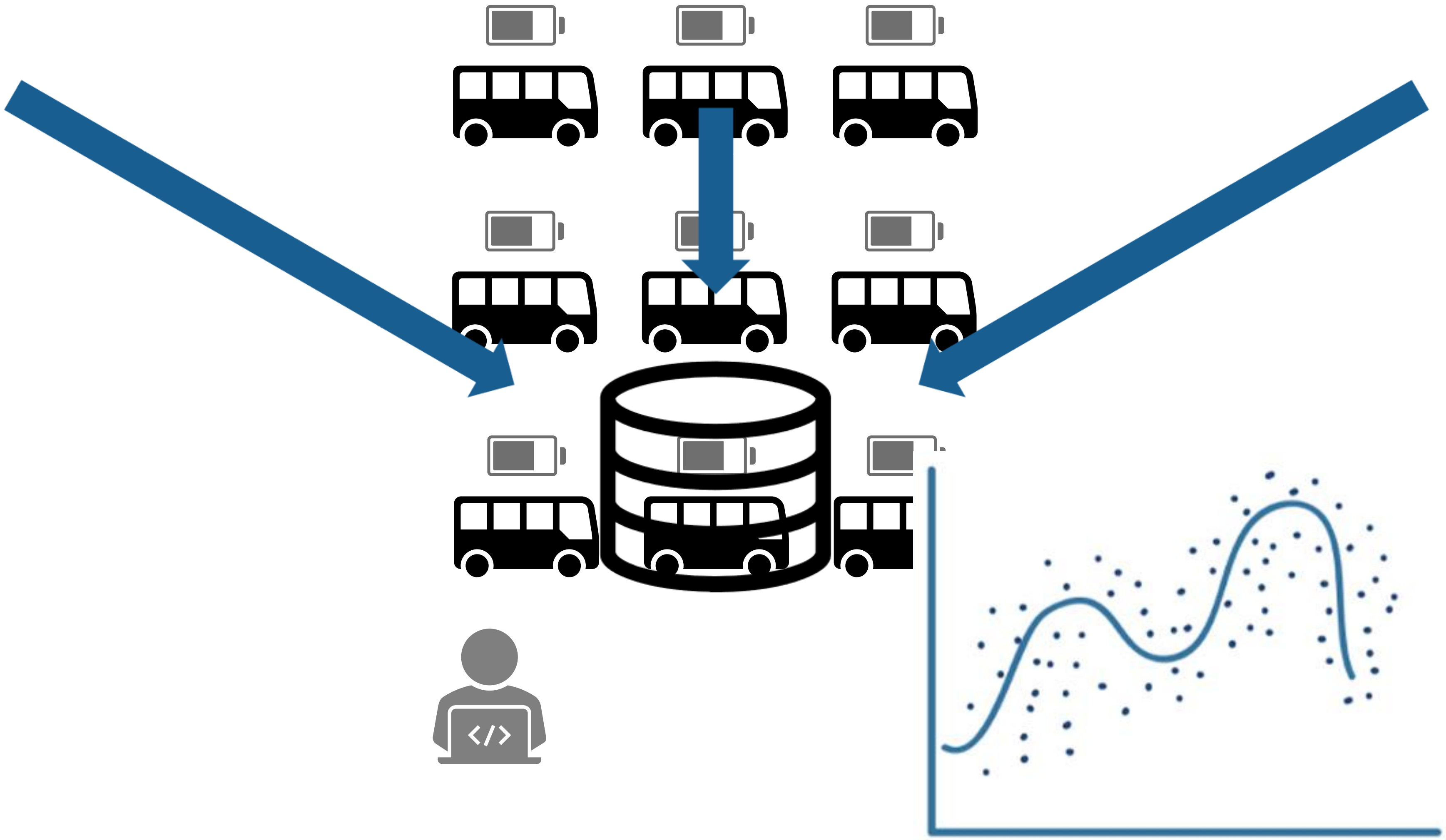
Drift Detection



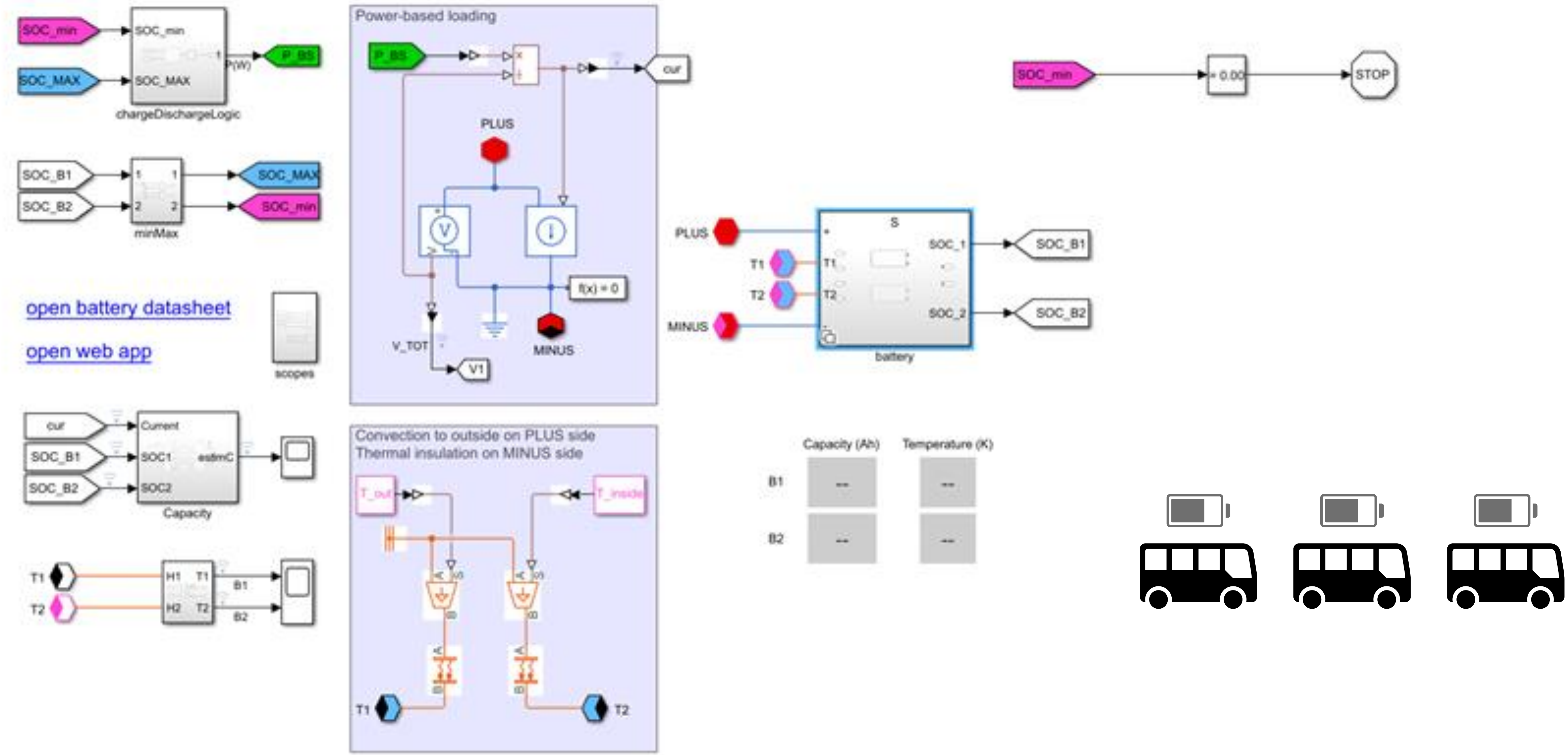
Let us remind ourselves the blueprint of the automated solution.



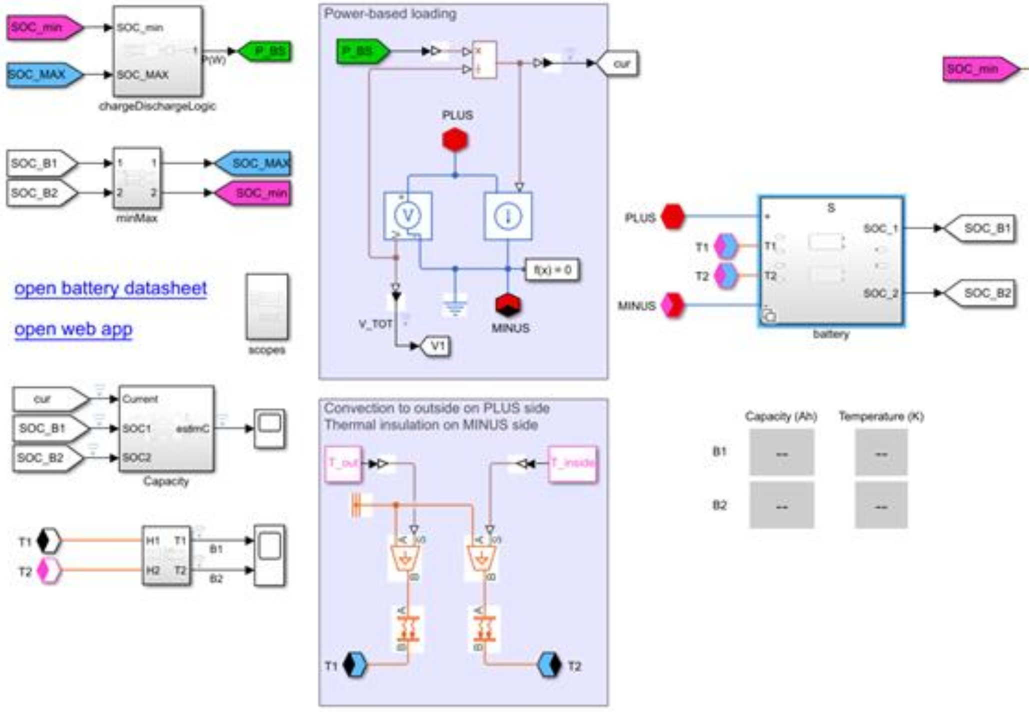
Data is everything for Machine Learning.

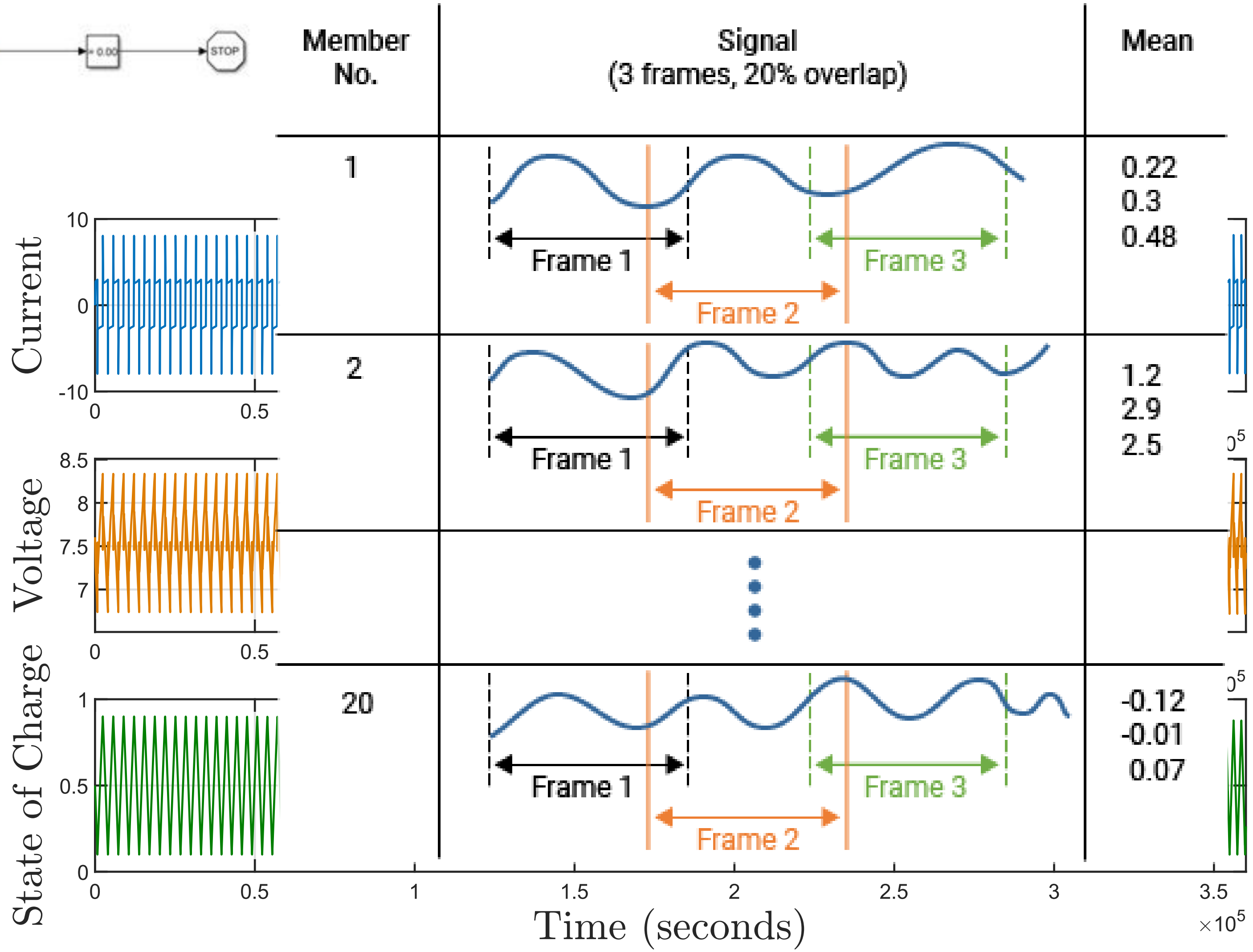


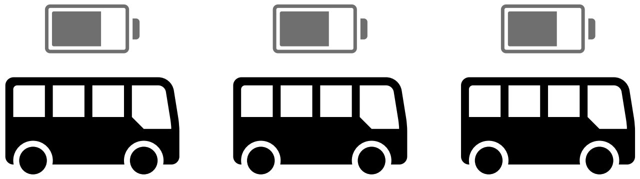
Physics-based simulation allows realistic data generation.



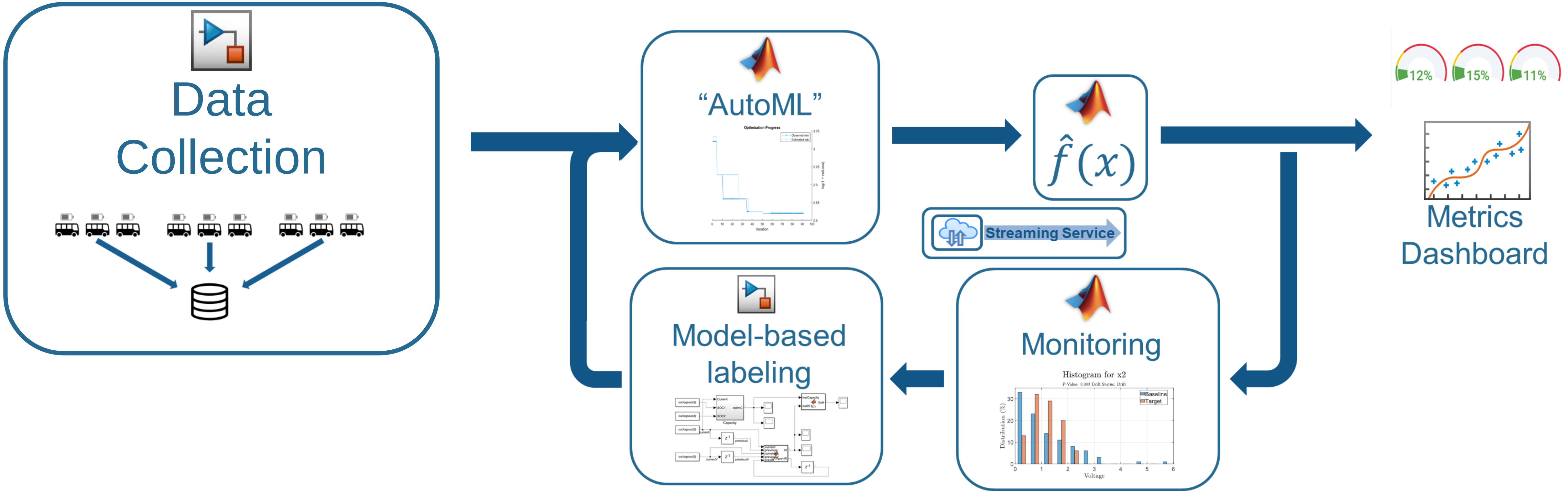
Physics-based simulation allows realistic data generation.



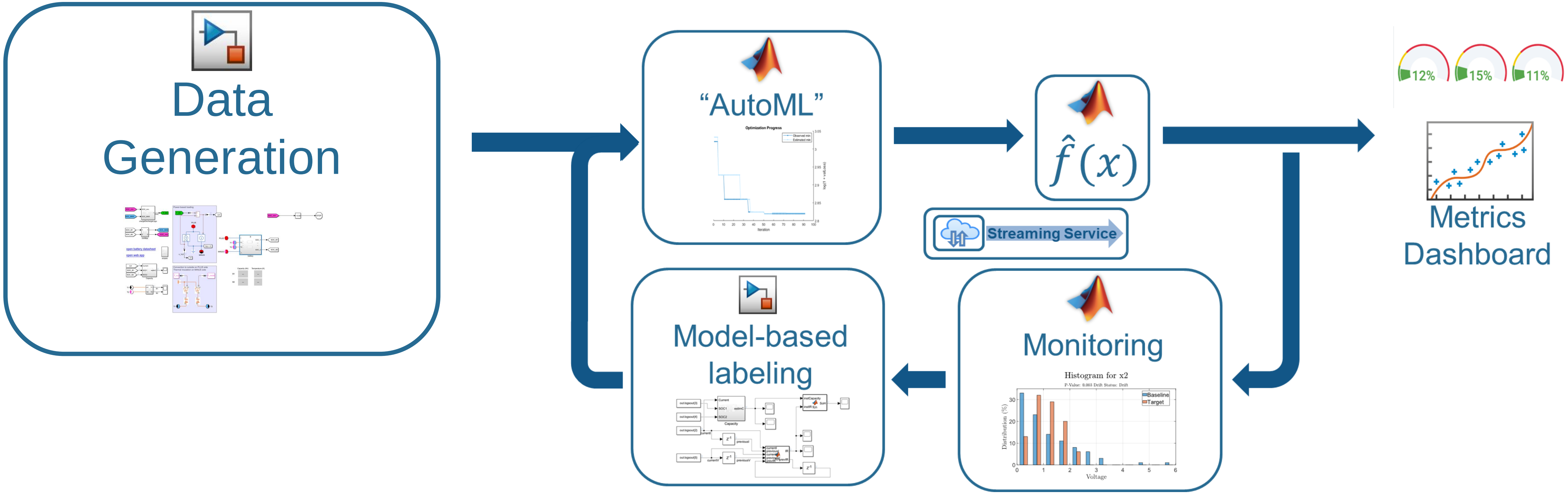




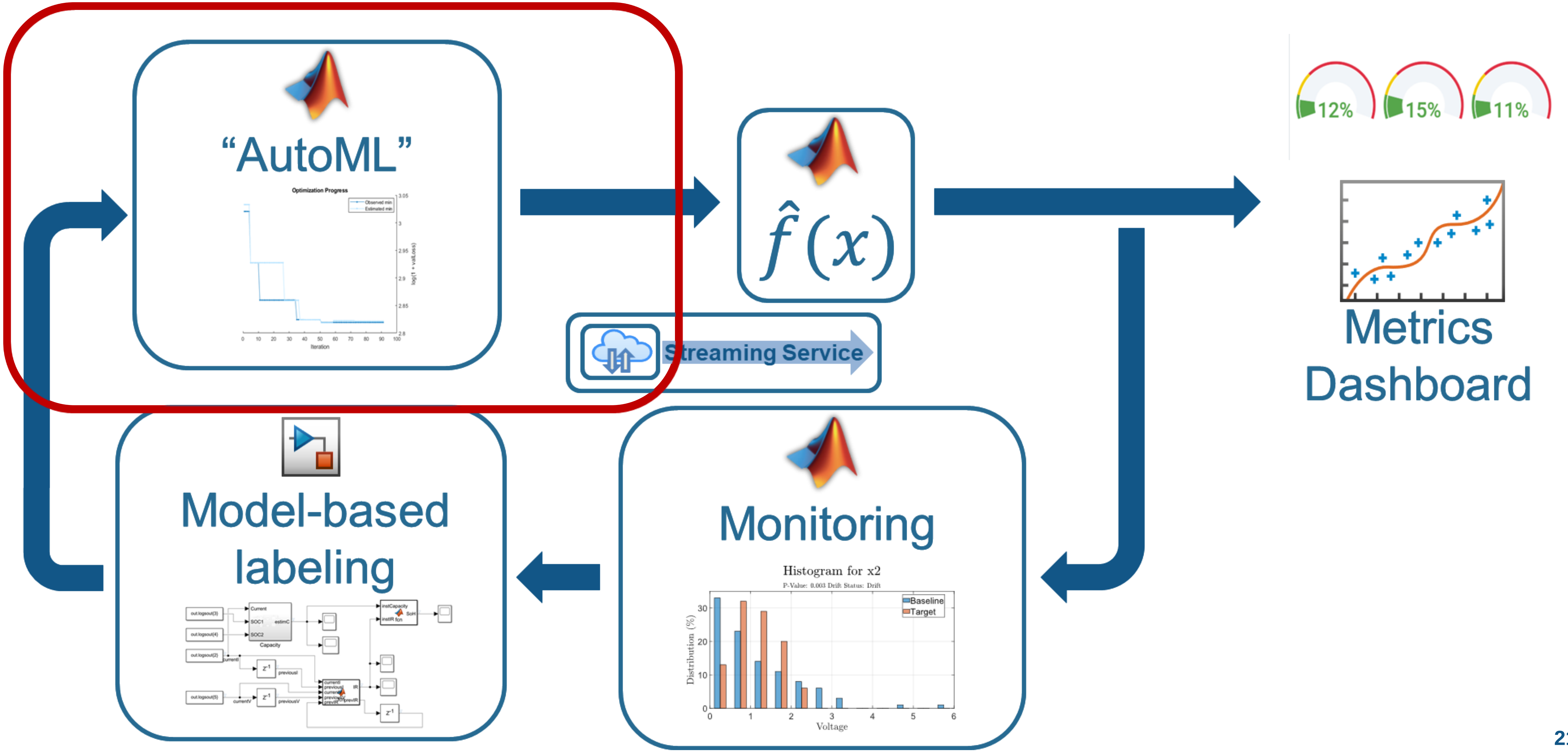
Physics-based simulation allows realistic data generation.



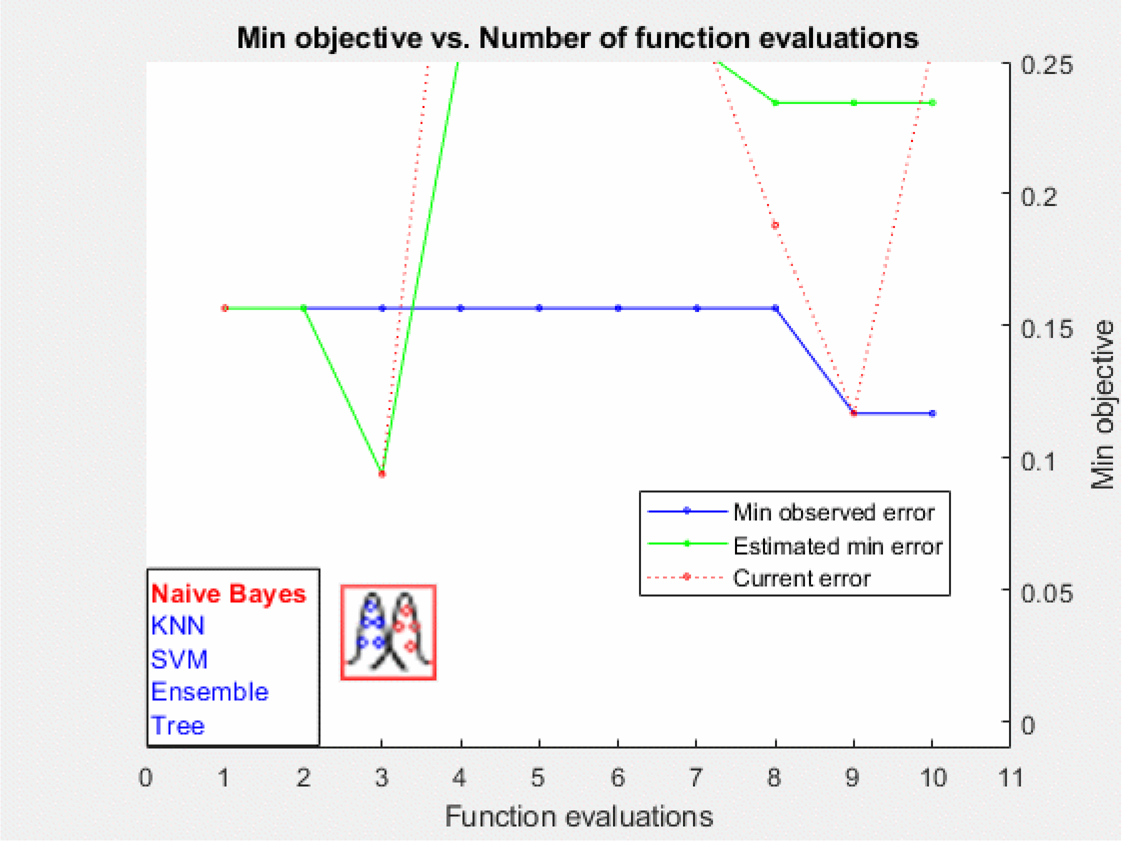
Physics-based simulation allows realistic data generation.

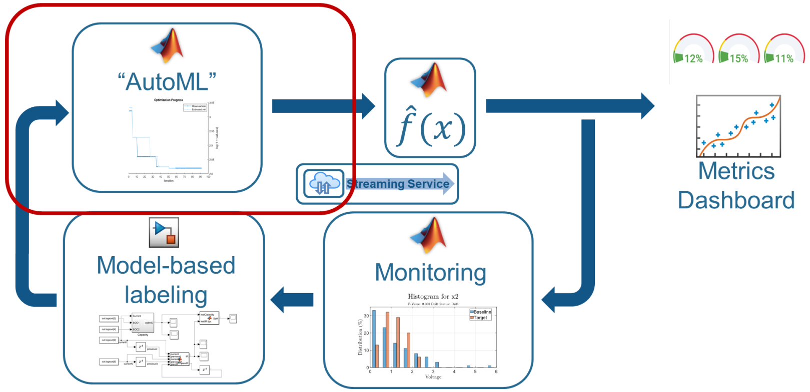


AutoML paves the way for automated training of data-driven algorithms.

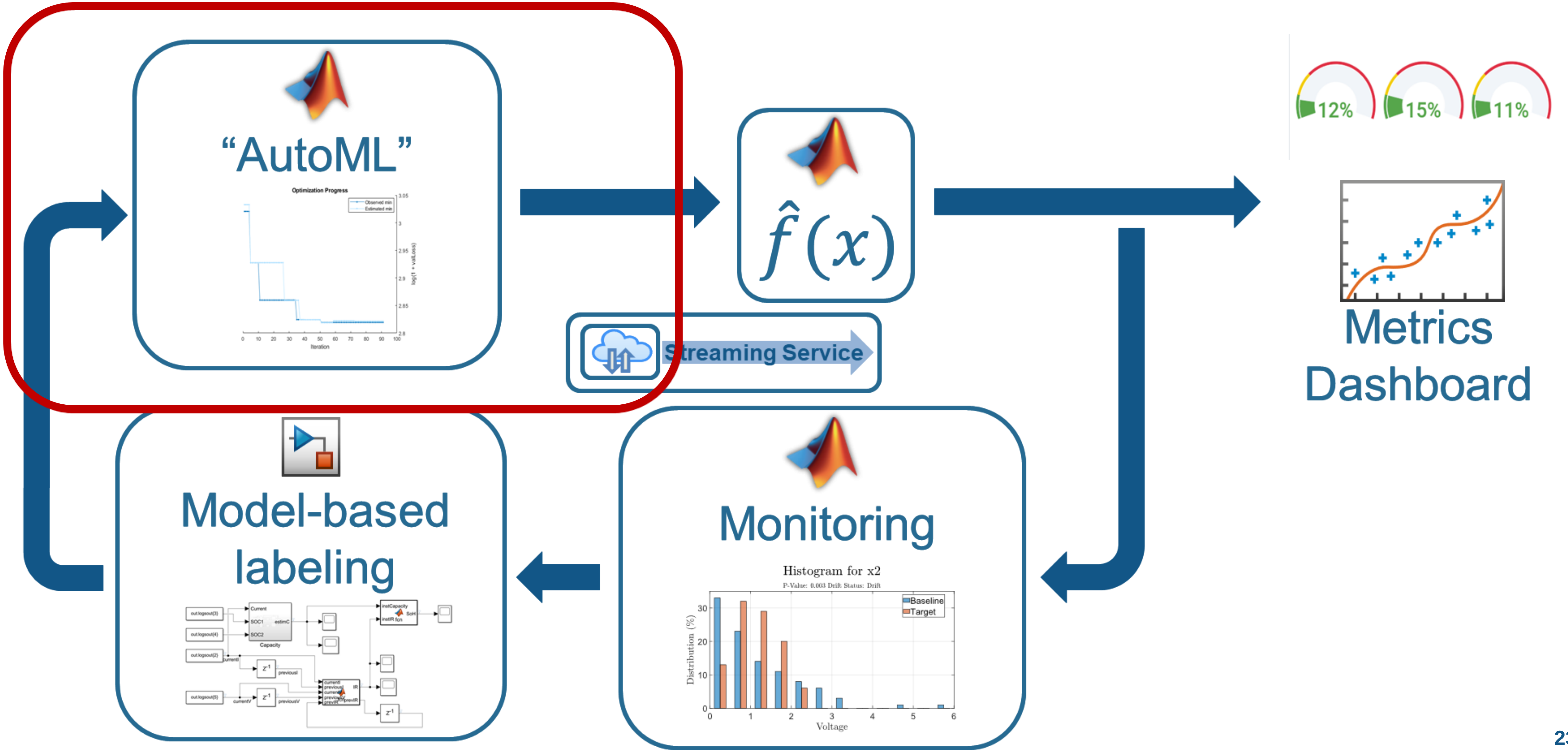


AutoML “automagically” finds the right model.

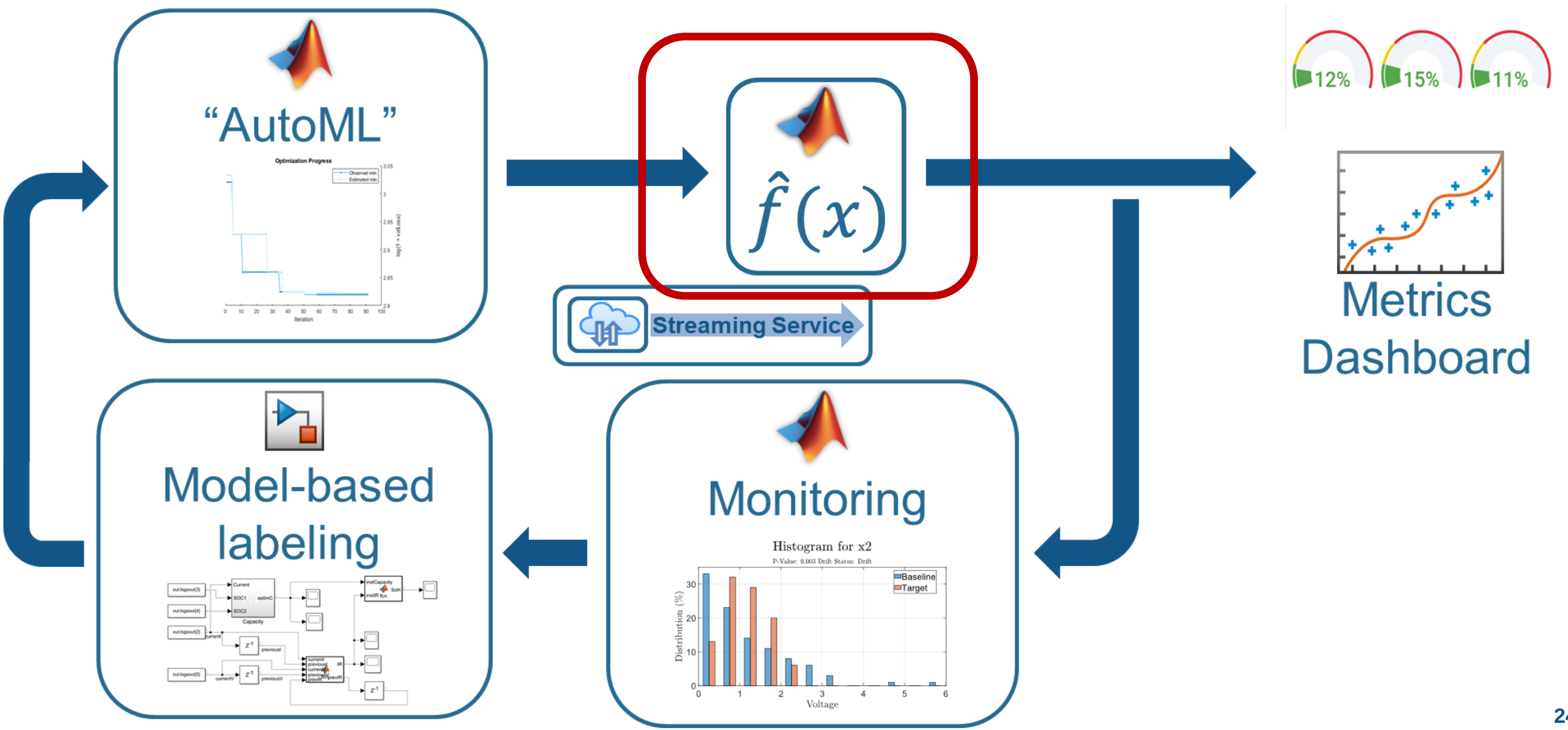




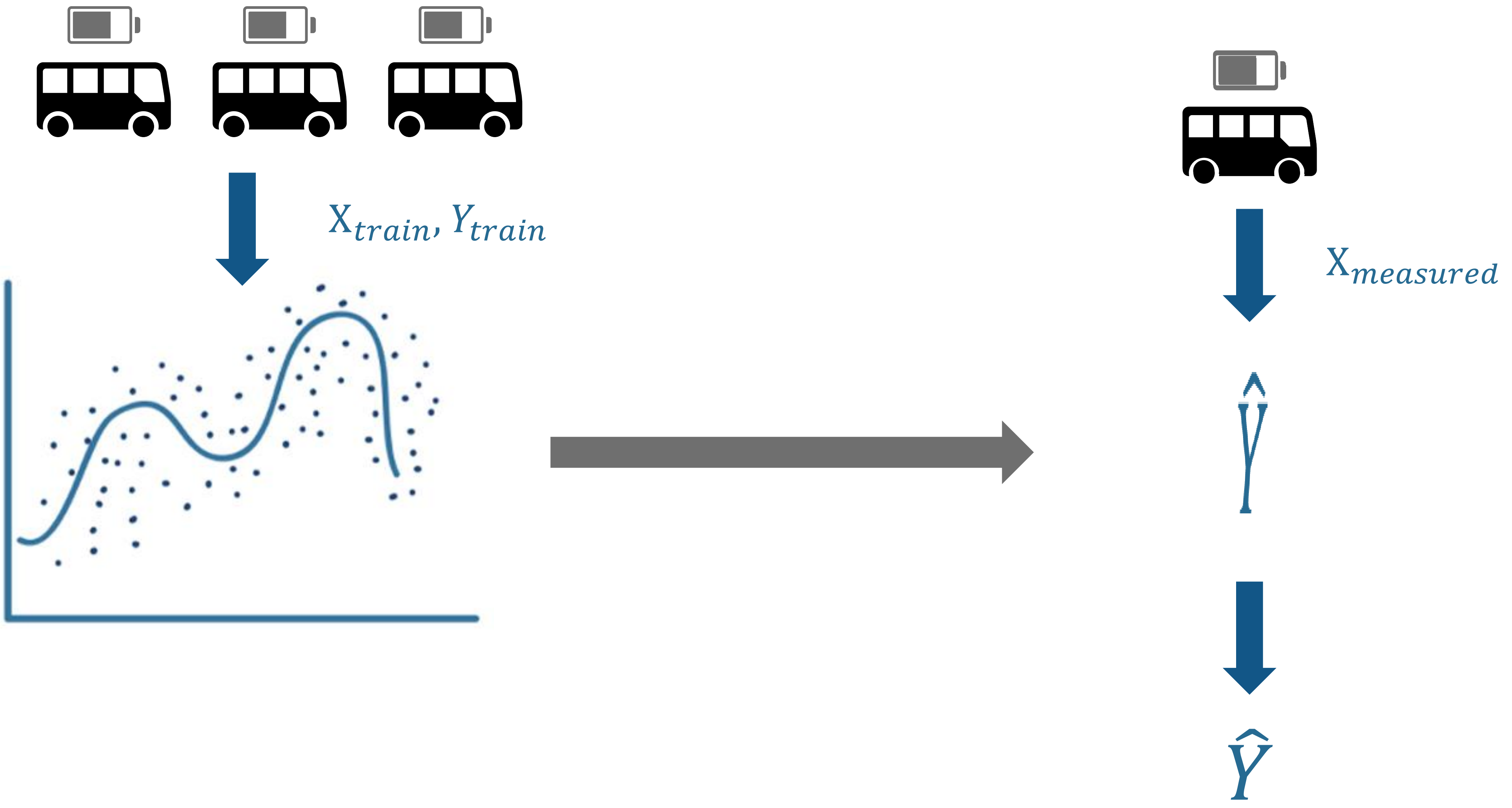
AutoML paves the way for automated training of data-driven algorithms.



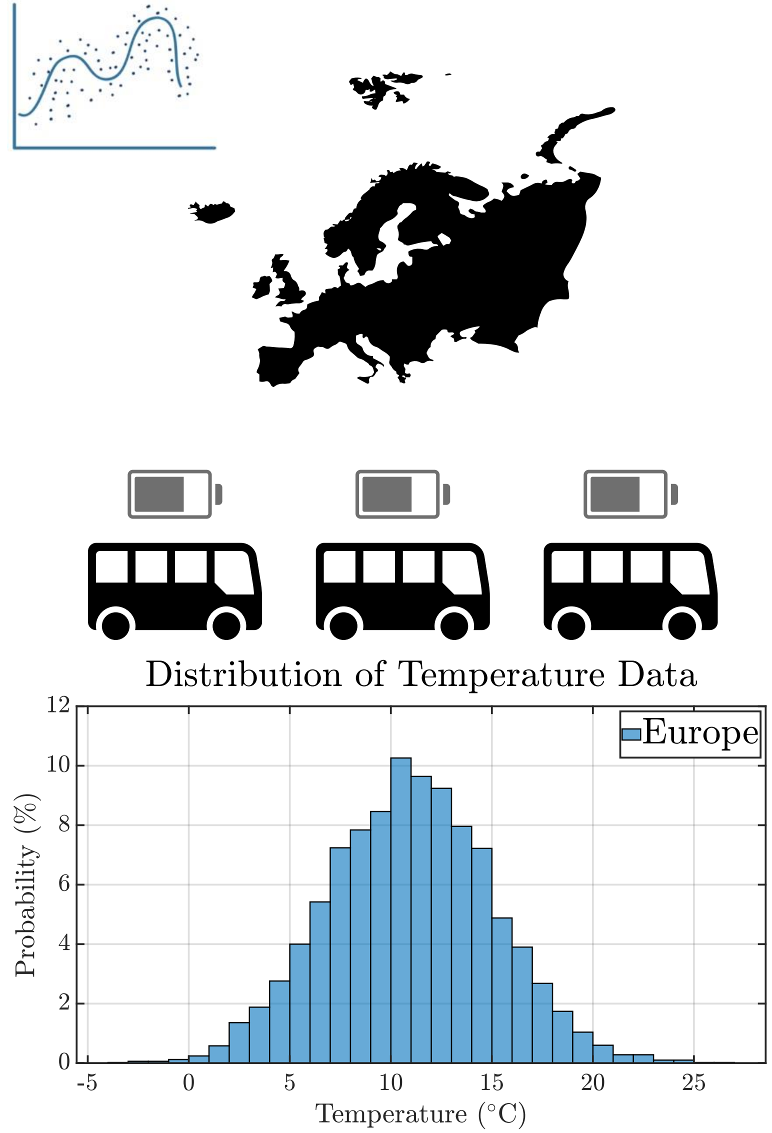
AutoML paves the way for automated training of data-driven algorithms.

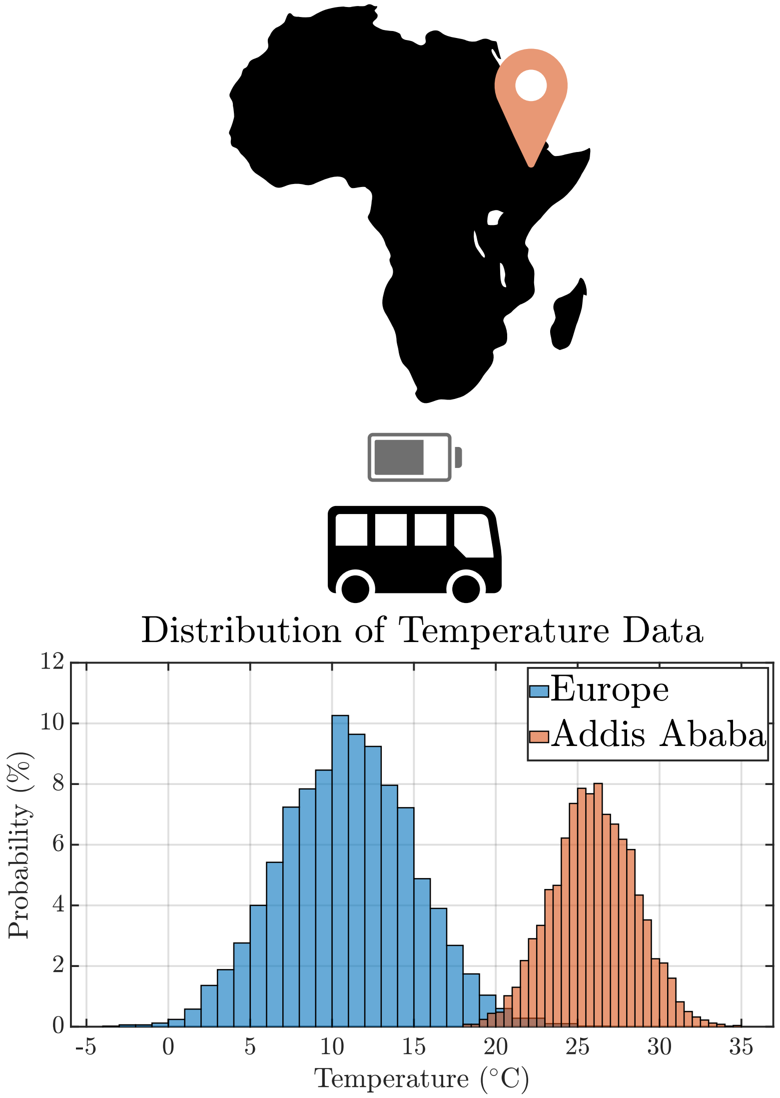


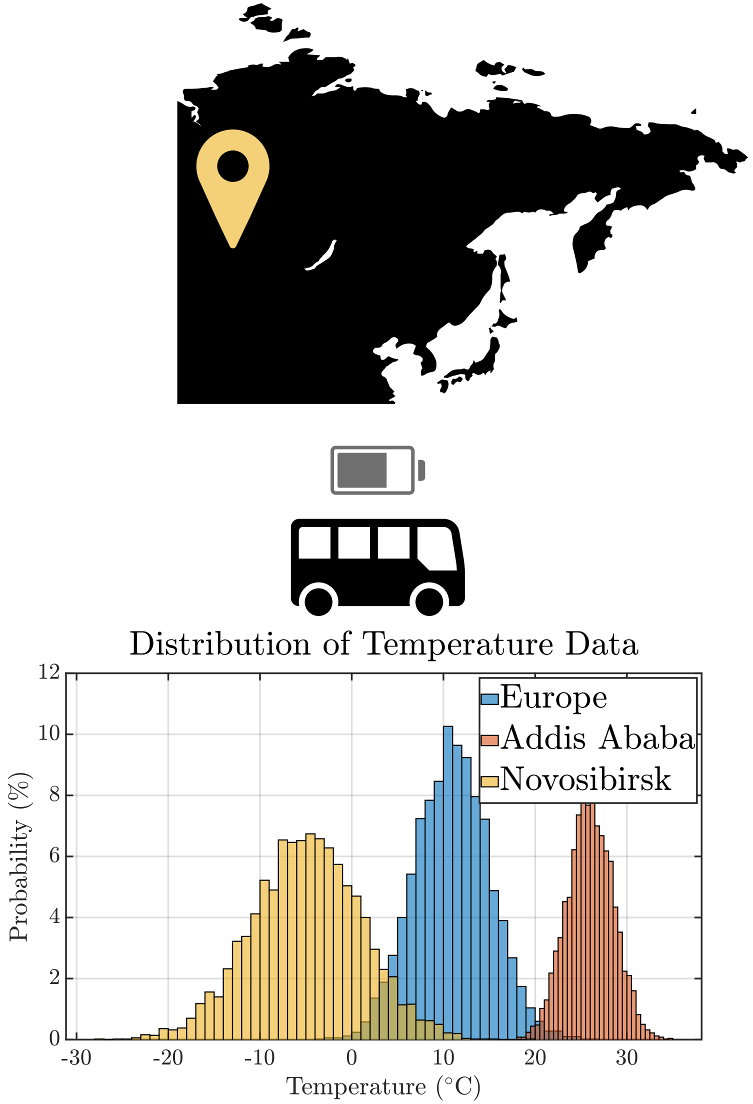
Machine Learning models generally assume training data is static.



Static data assumption rarely holds in the real world.







Detecting concept drift is challenging, detecting data drift is easier and practical.

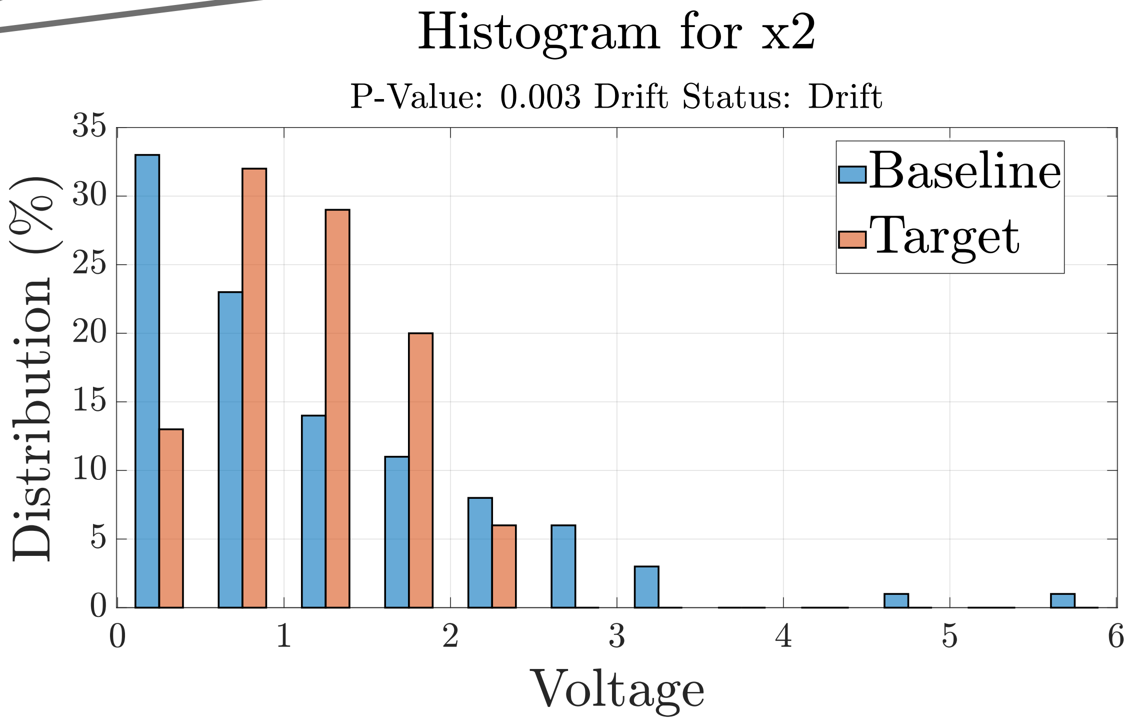
Baseline

Target

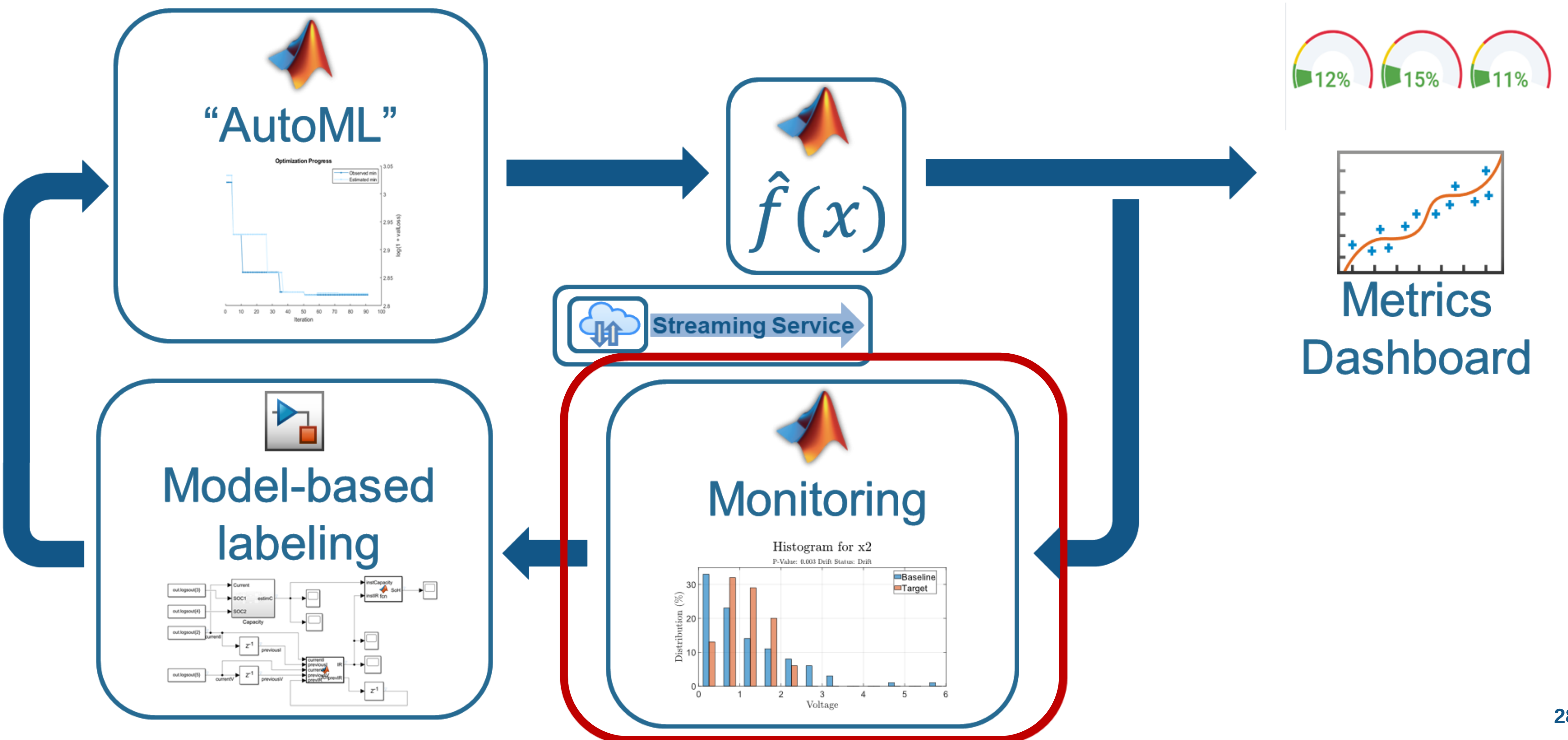
$$\exists\, t:\ P_t(\boxed{X}, Y) \neq P_{t+1}(\boxed{X}, Y)$$

Features

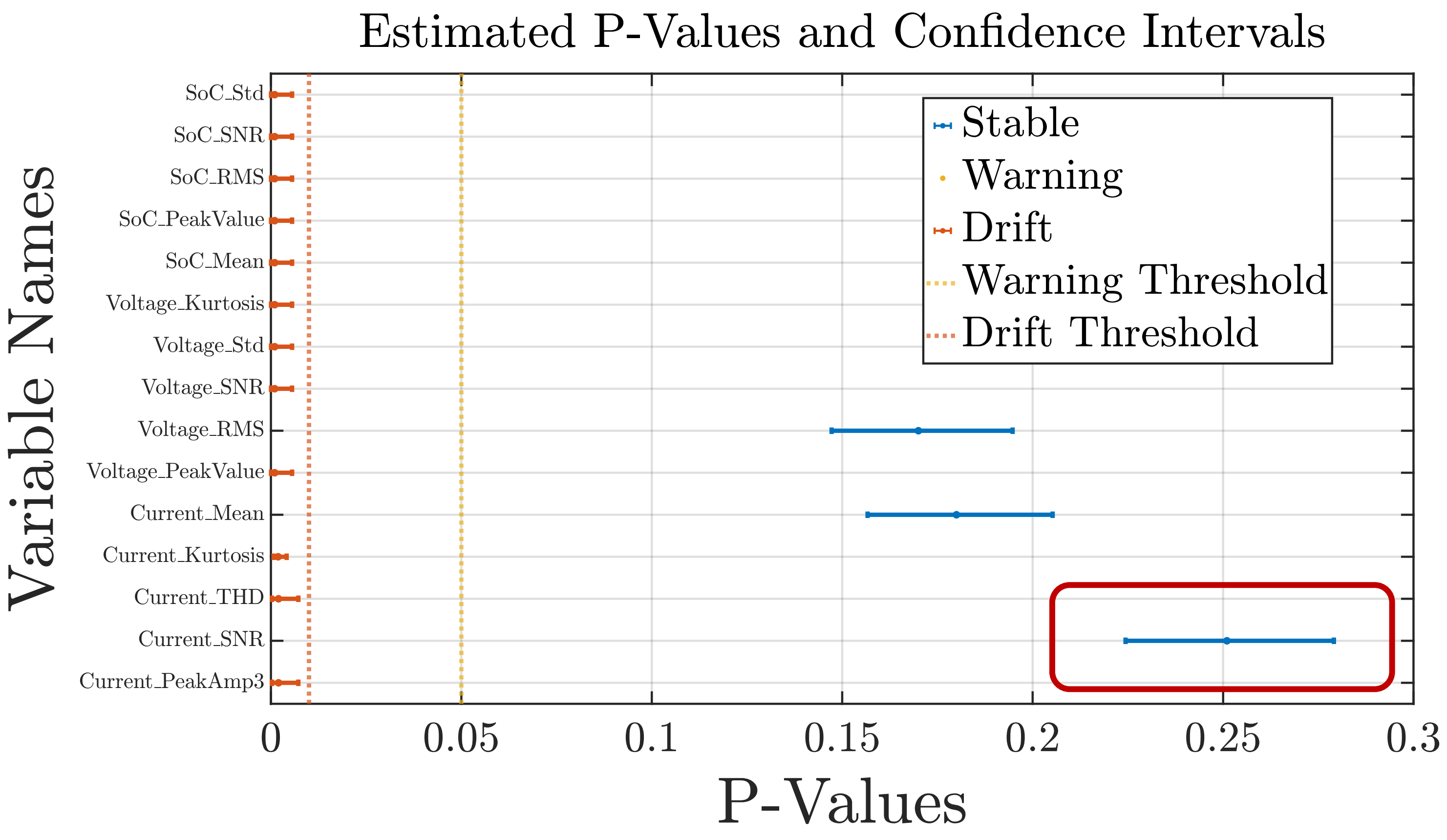
Labels/Responses

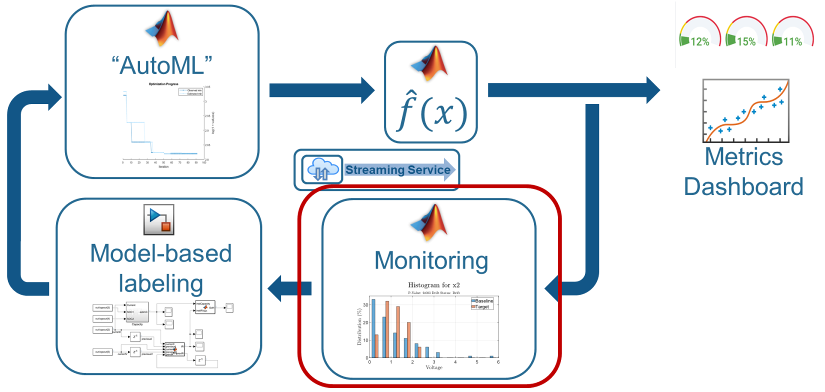


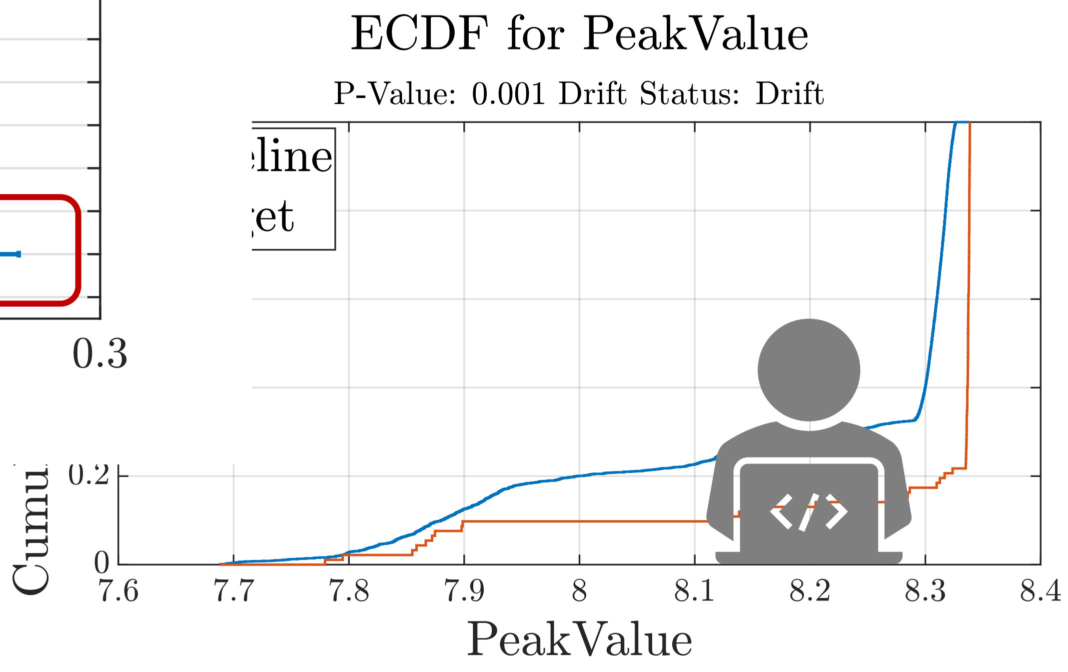
Drift monitoring periodically checks for and detects changes in data.



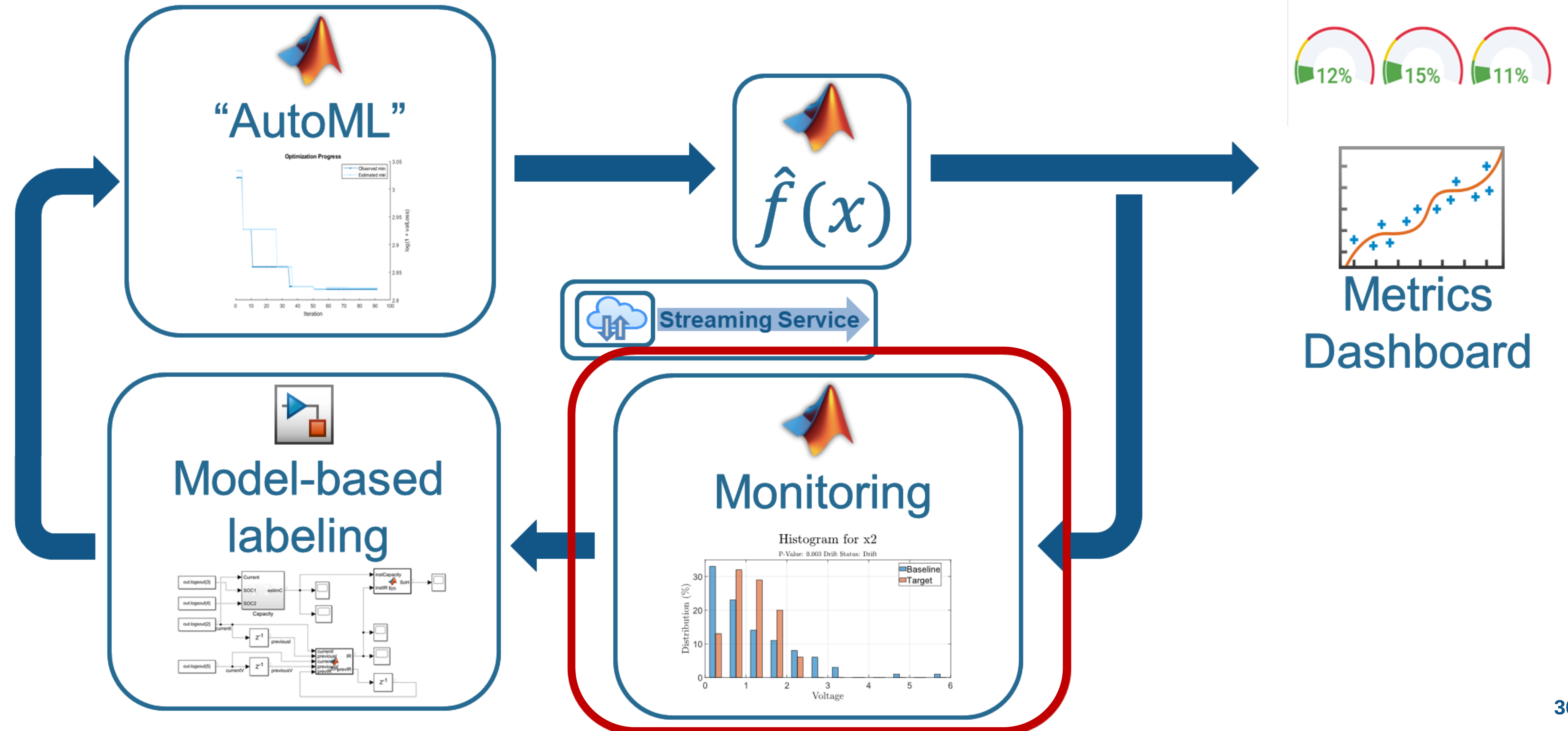
Per-feature drift can be visualized, interpreted and assessed in an automated way or with human supervision.



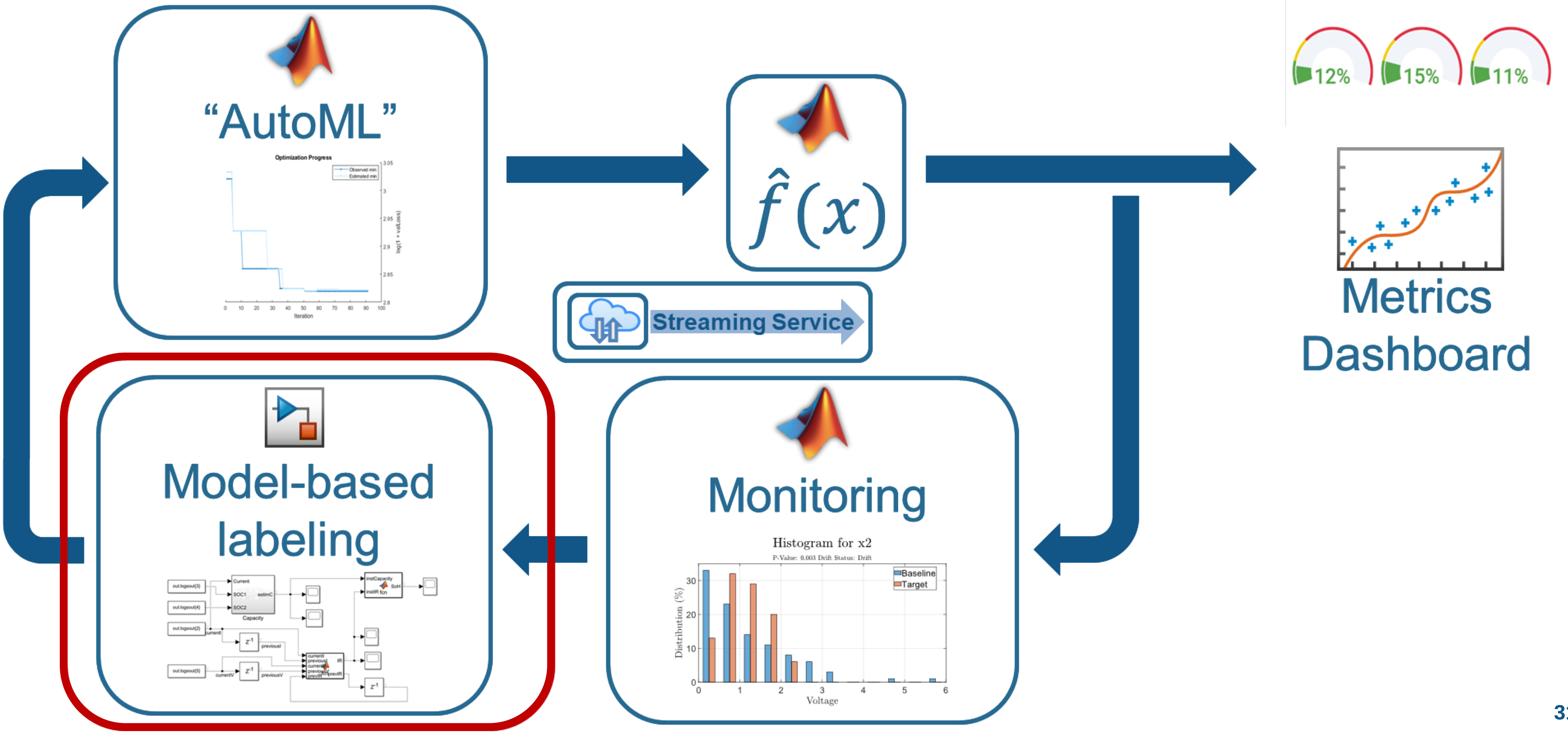




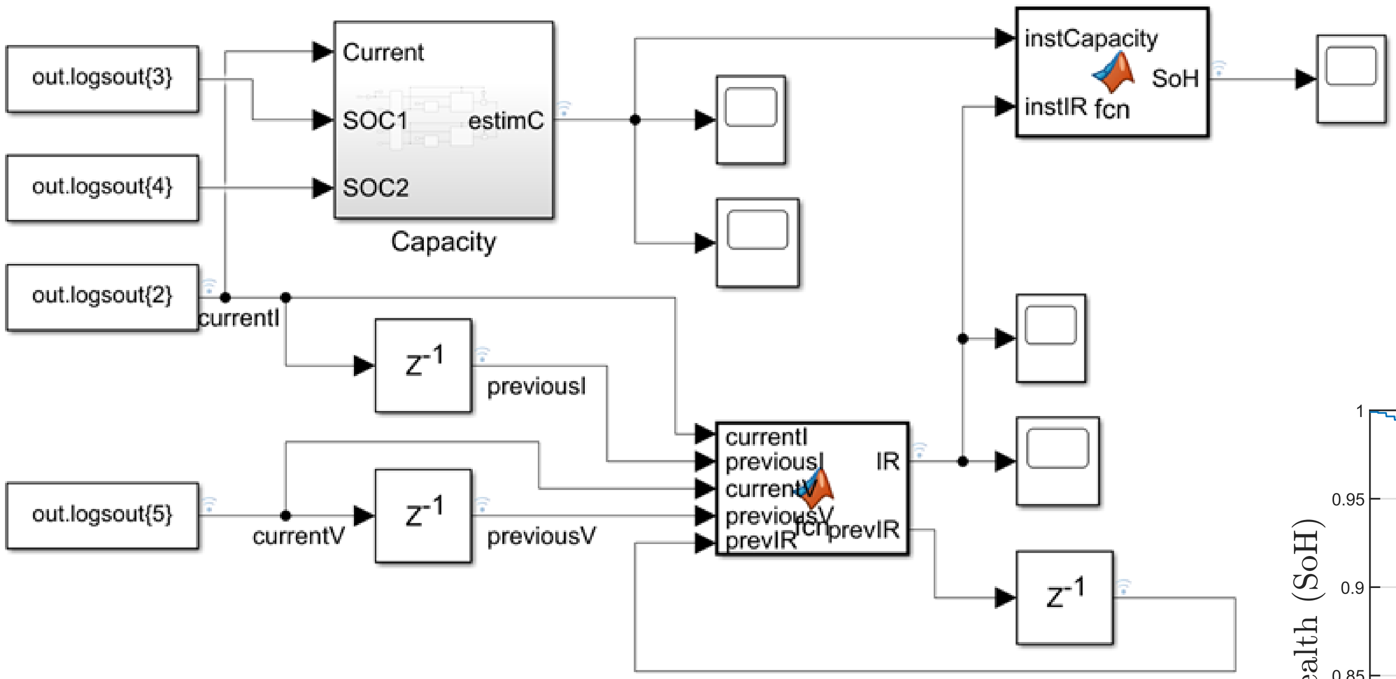
Drift monitoring system enhances automated solution with the ability to forecast when models may require retraining.

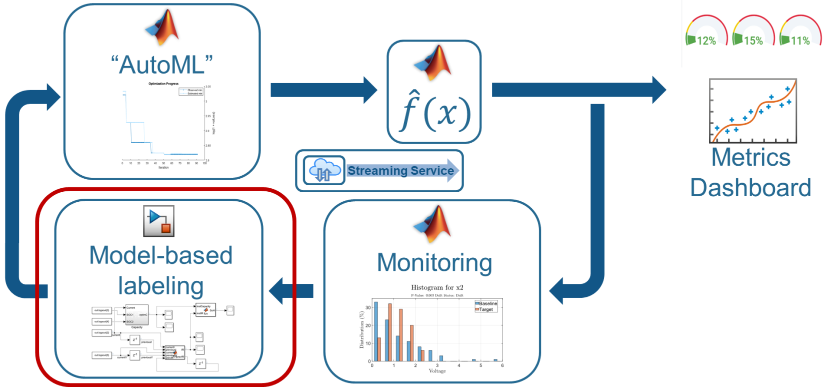


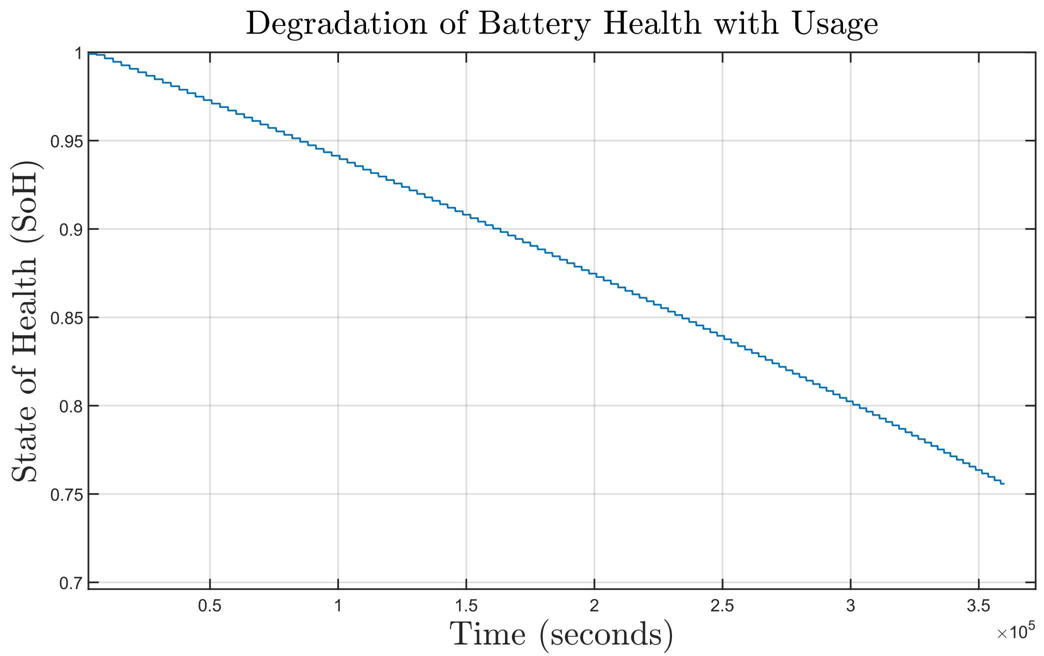
Drift monitoring system enhances automated solution with the ability to forecast when models may require retraining.



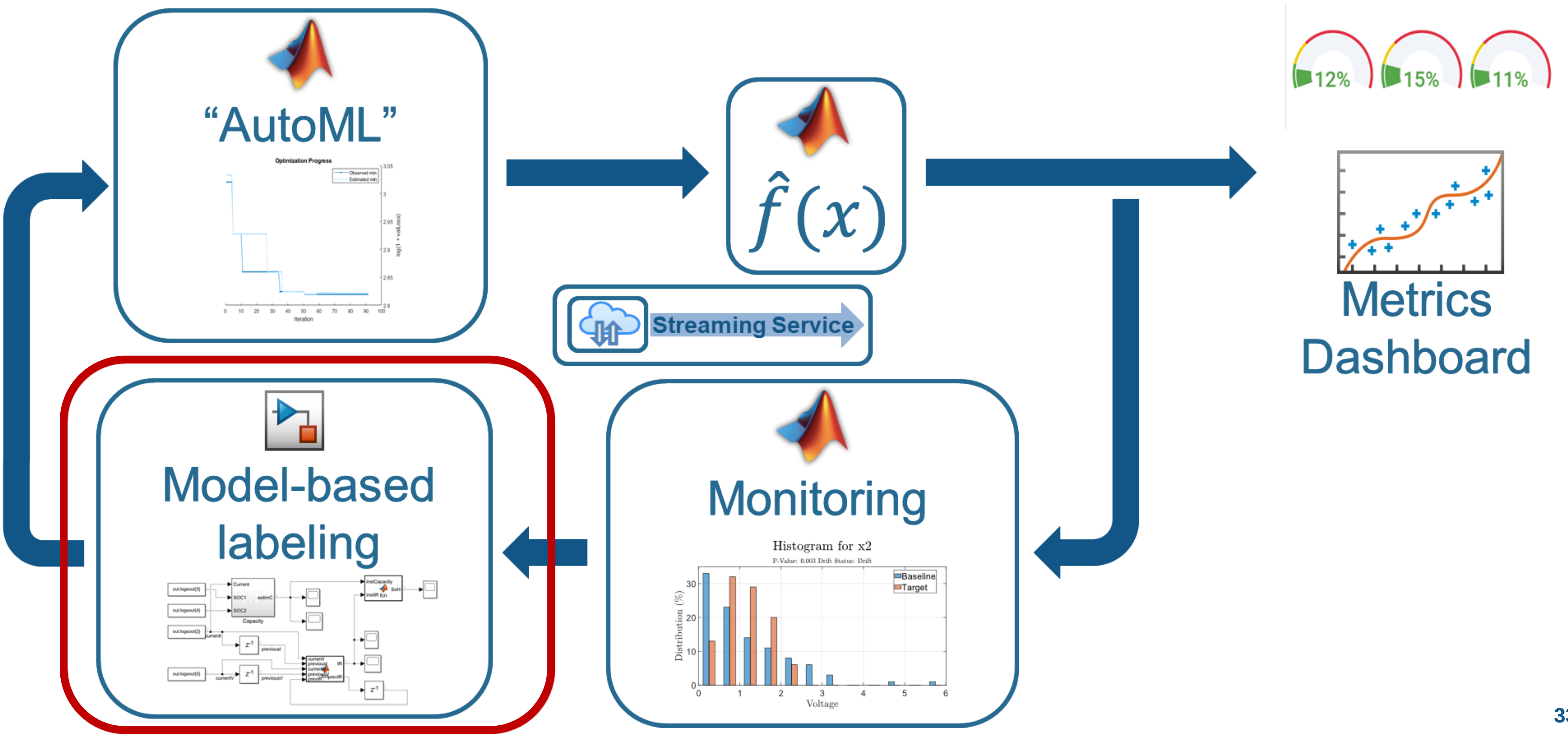
Model-based labeling system is high fidelity, but slow. Used for labeling only when prompted.



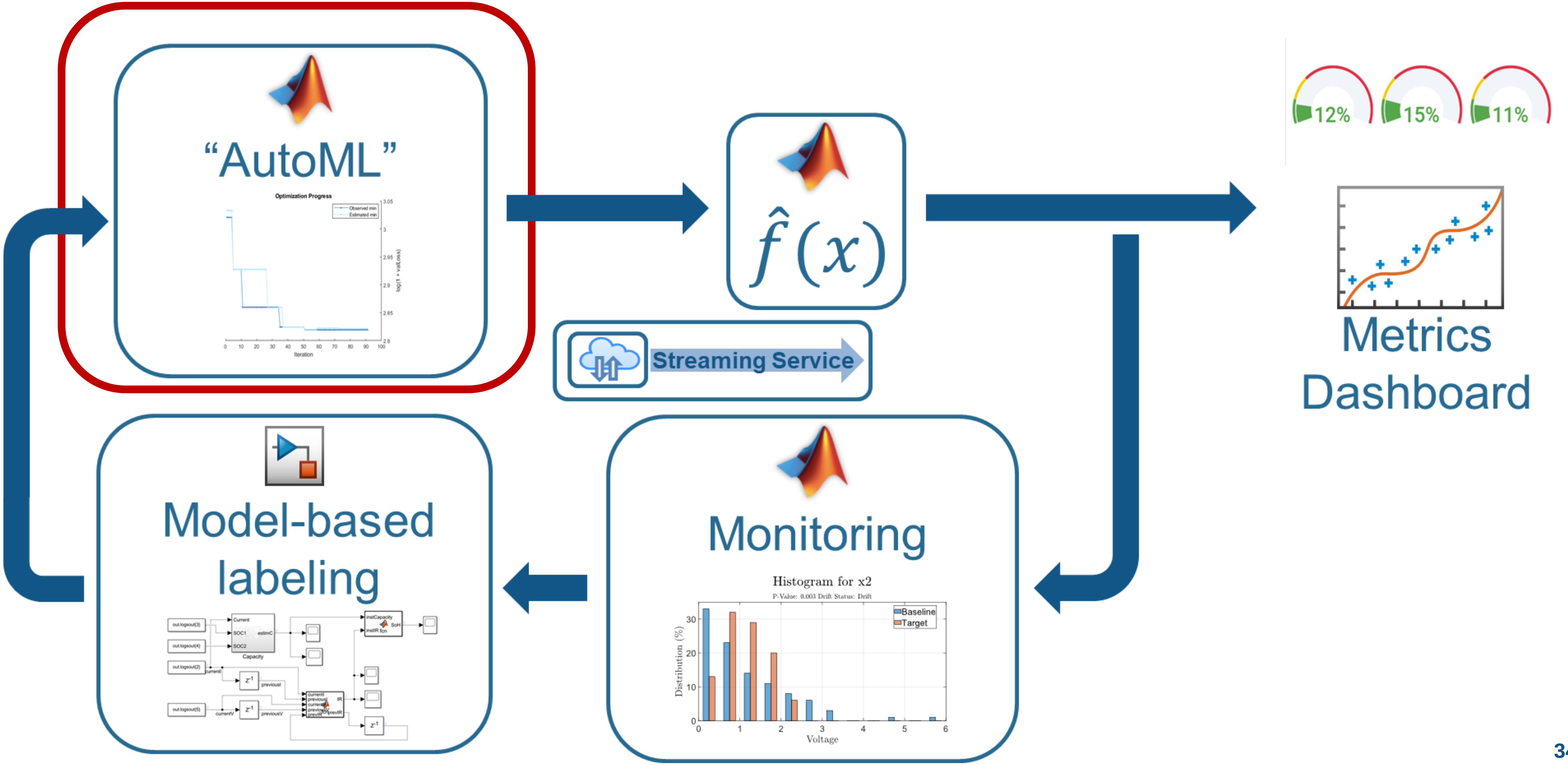




Drift resulting in model performance degradation triggers retraining.



Drift resulting in model performance degradation triggers retraining.



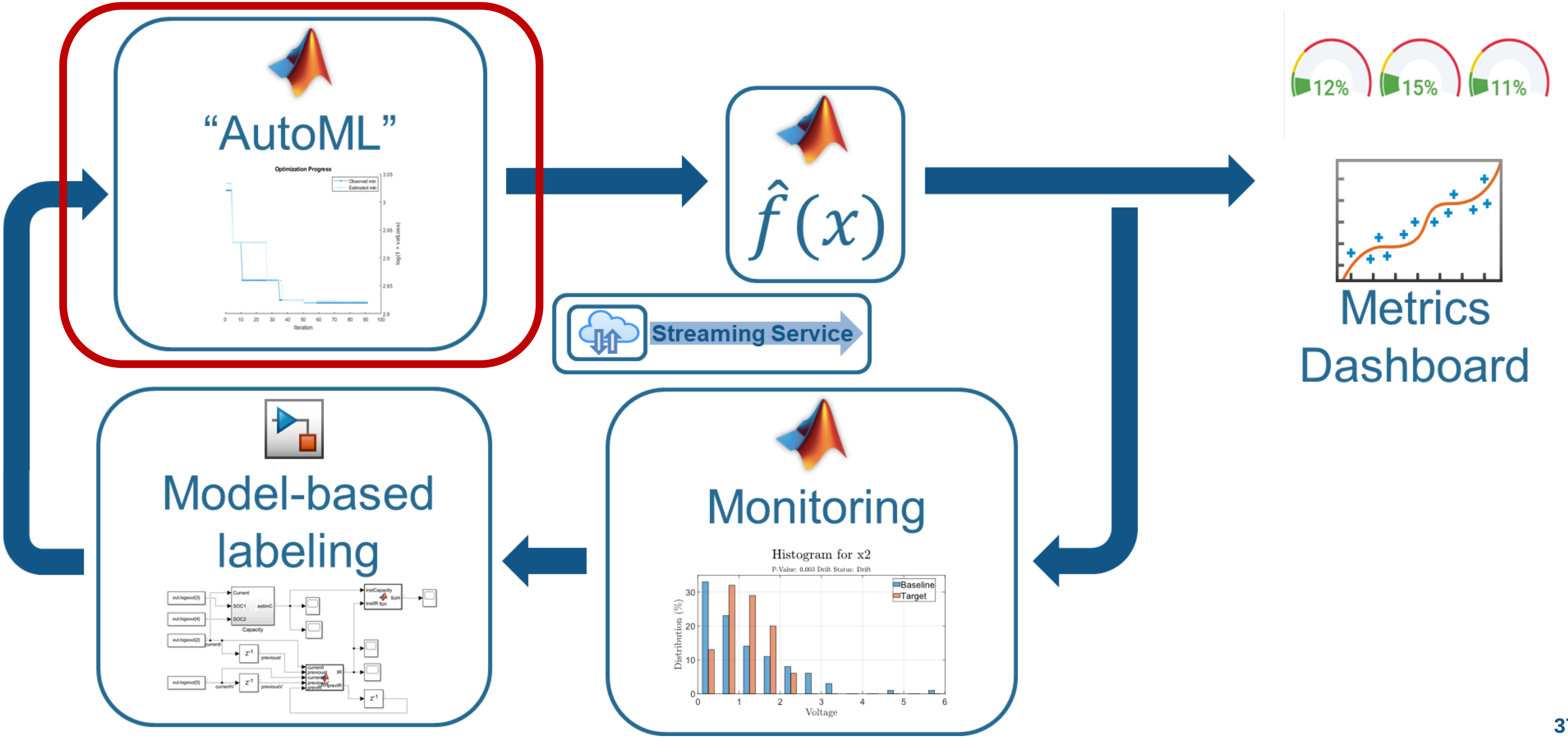
Model in production is replaced if challenger model has better performance.

$$\mathrm{MSE}\left(\boxed{\hat{f}_{challenger}(x)}, Y\right) \stackrel{?}{<} \mathrm{MSE}\left(\boxed{\hat{f}_{champion}(x)}, Y\right)$$

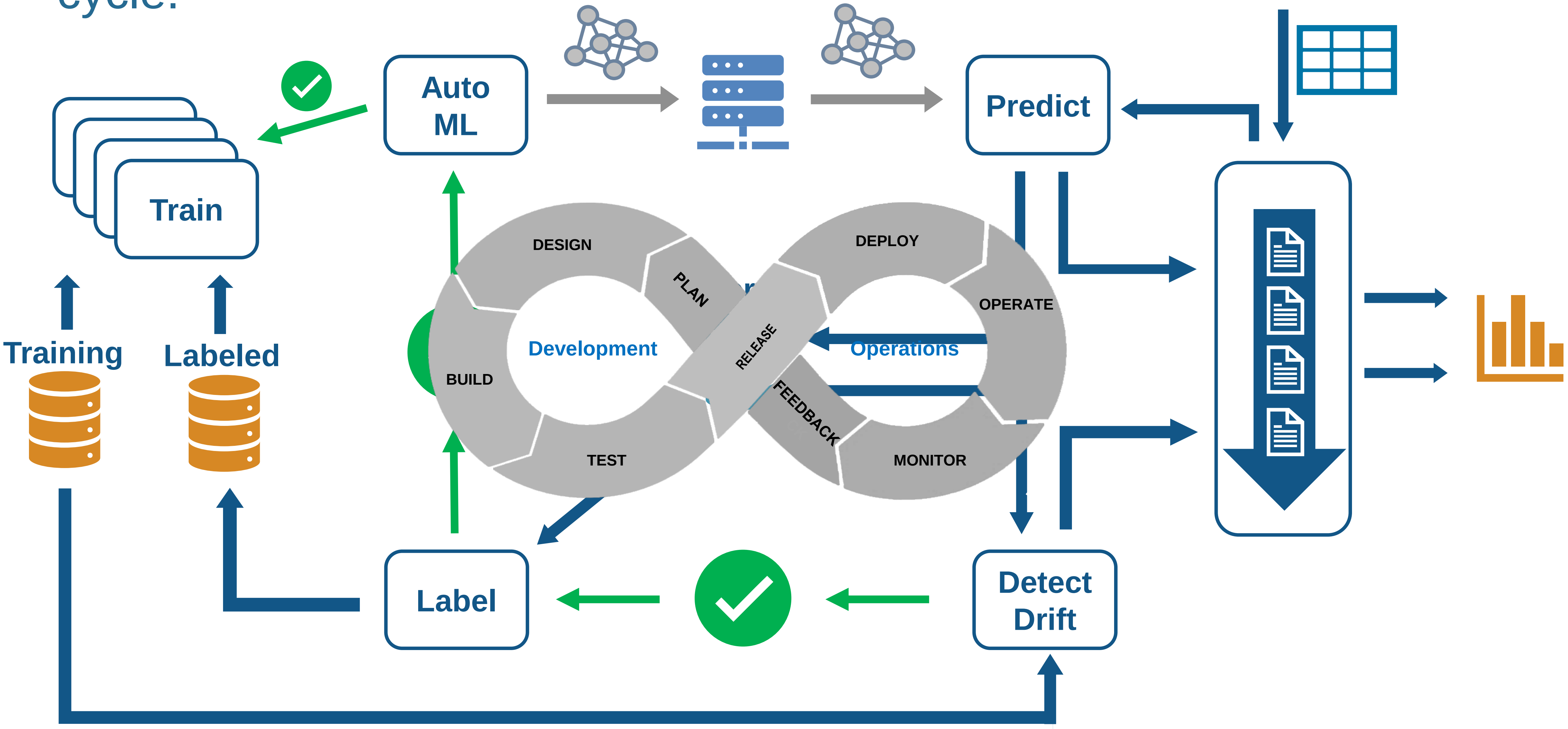
Model in production is replaced if challenger model has better performance.

$$\hat{f}_{champion}(x)$$

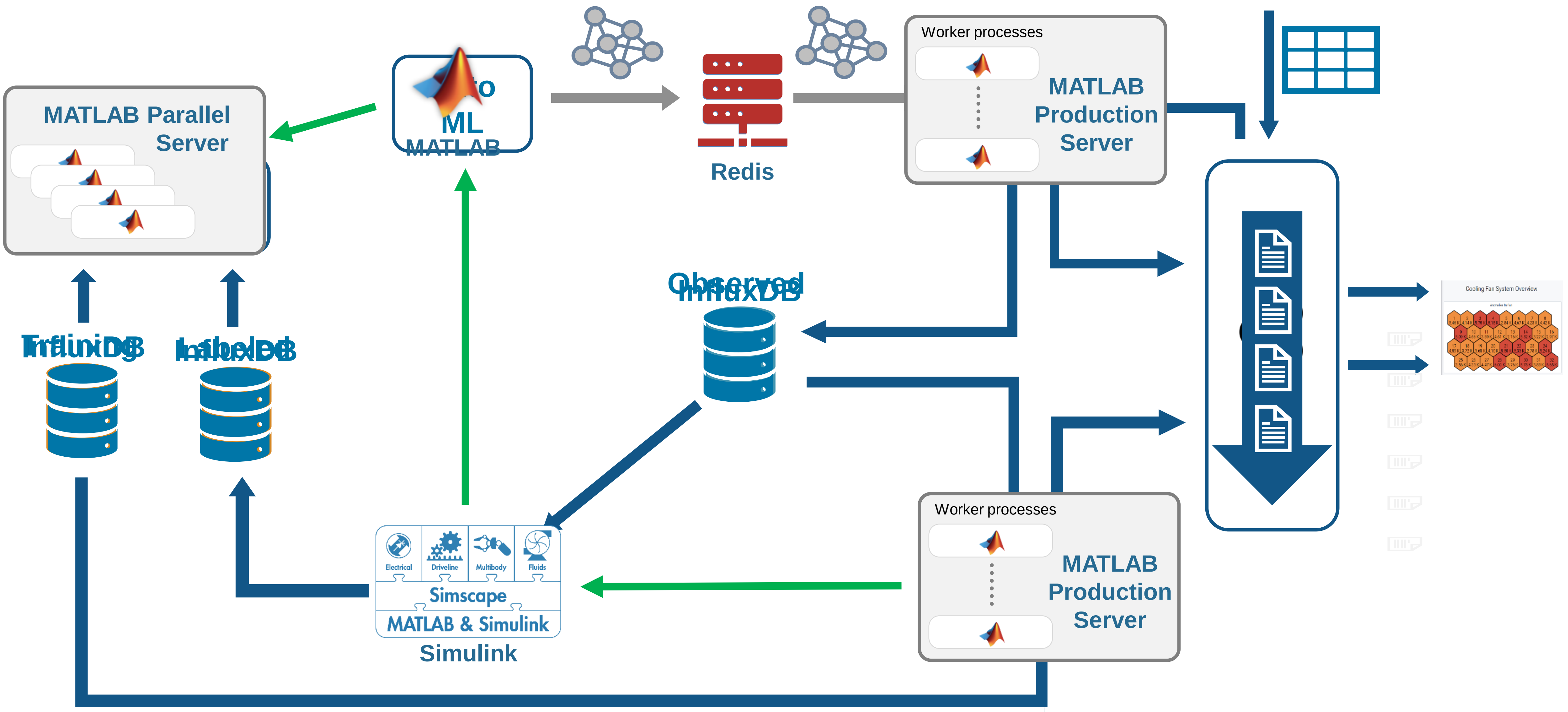
The train-deploy-monitor-label cycle automatically works on its own.



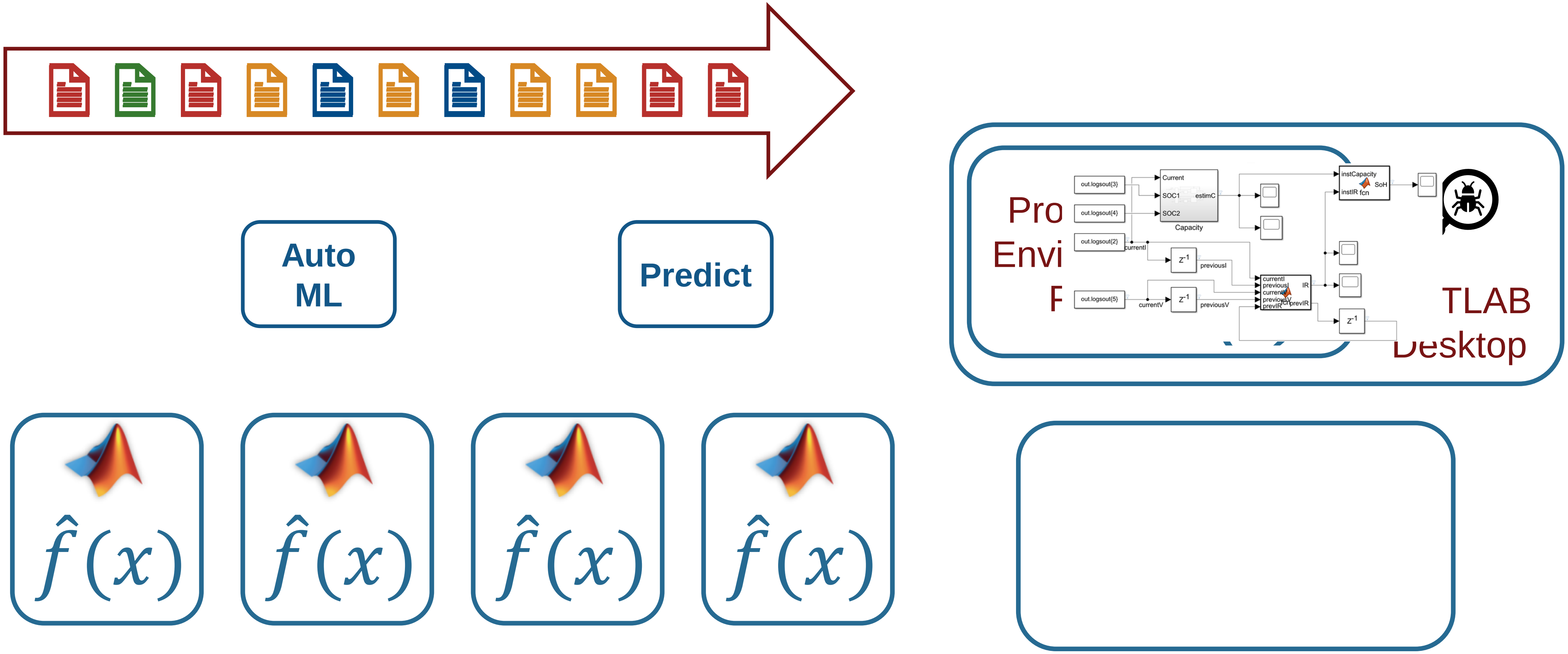
Production system architecture mirrors the stages of the Dev Ops cycle.



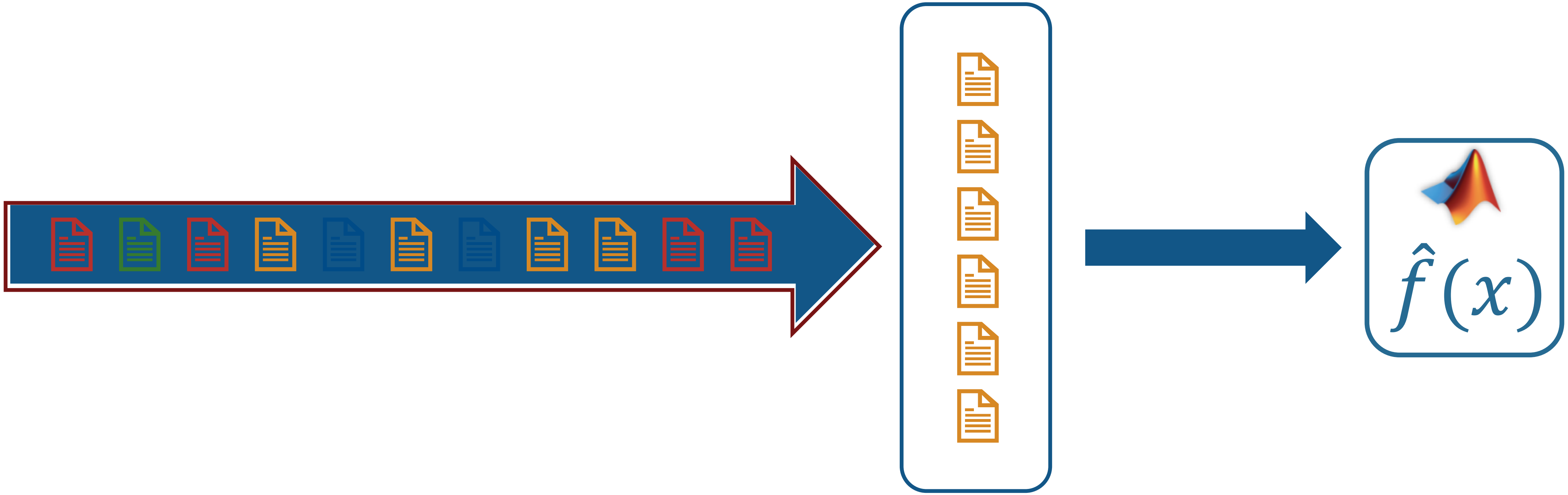
Off the shelf components minimize development effort.



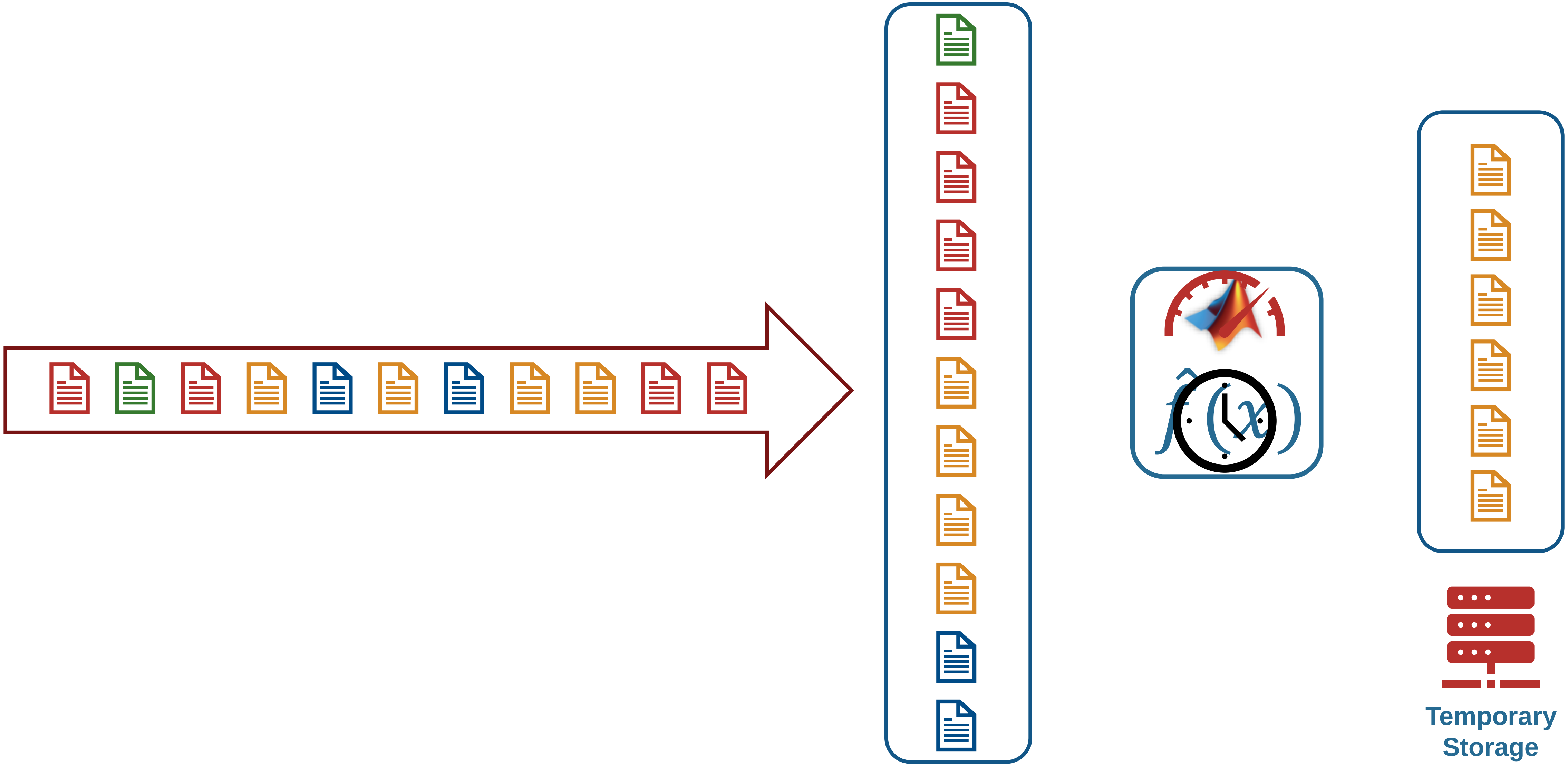
Data management drove the architecture. The development environment needs Dev Ops-specific features.



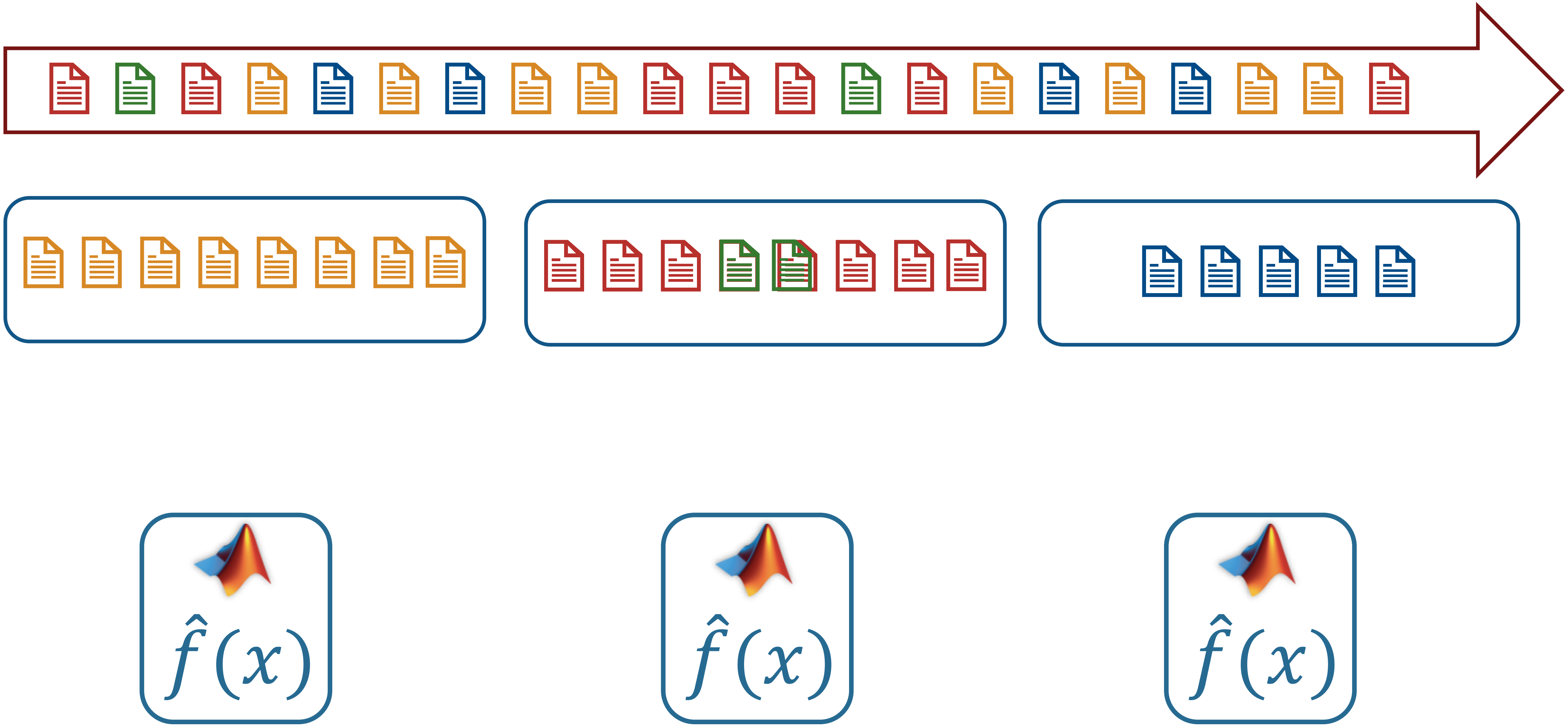
Prediction requires a complete frame of observations from a single battery, but the stream may not oblige.



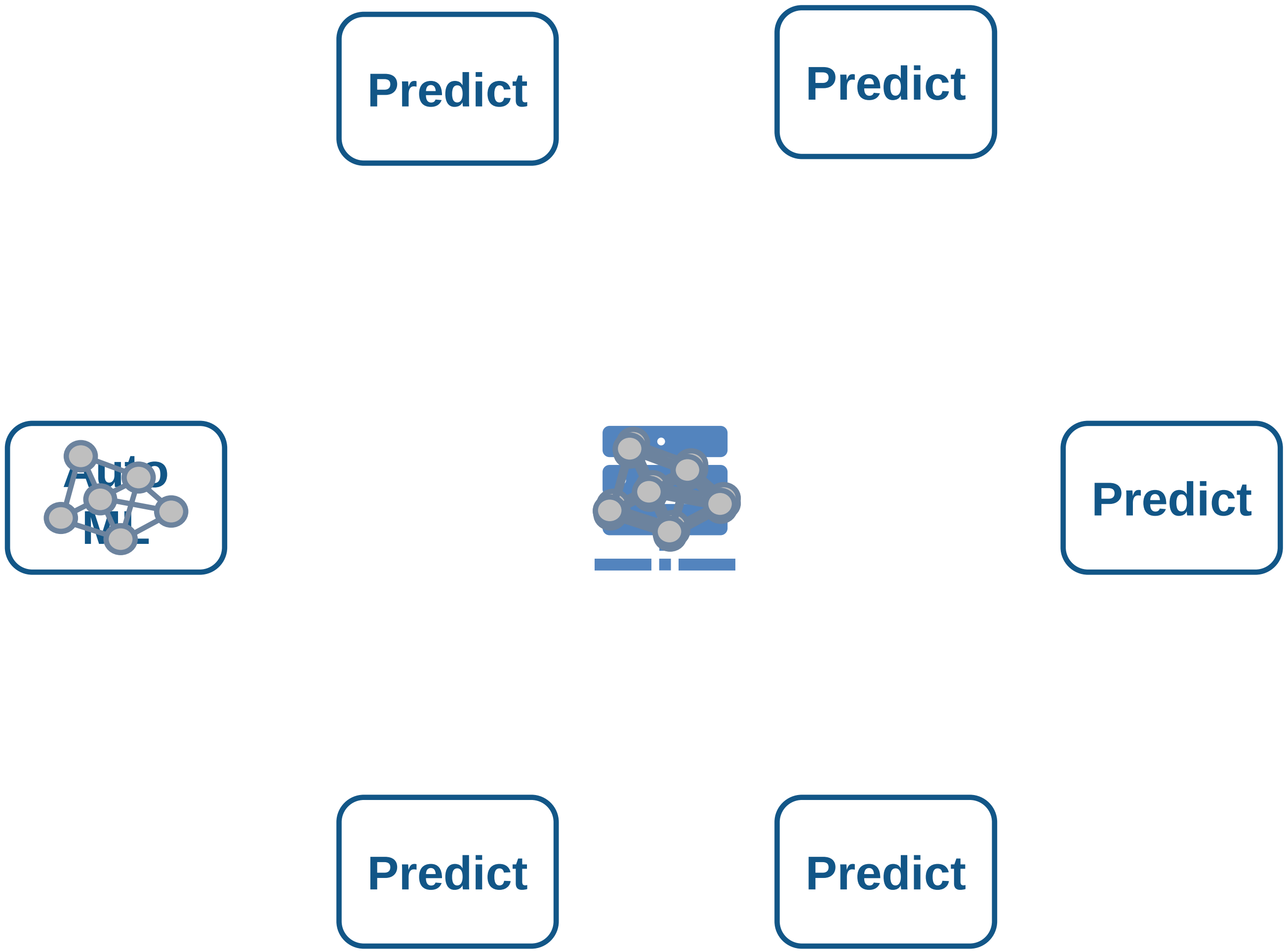
Build complete frames efficiently with message grouping and accumulation.



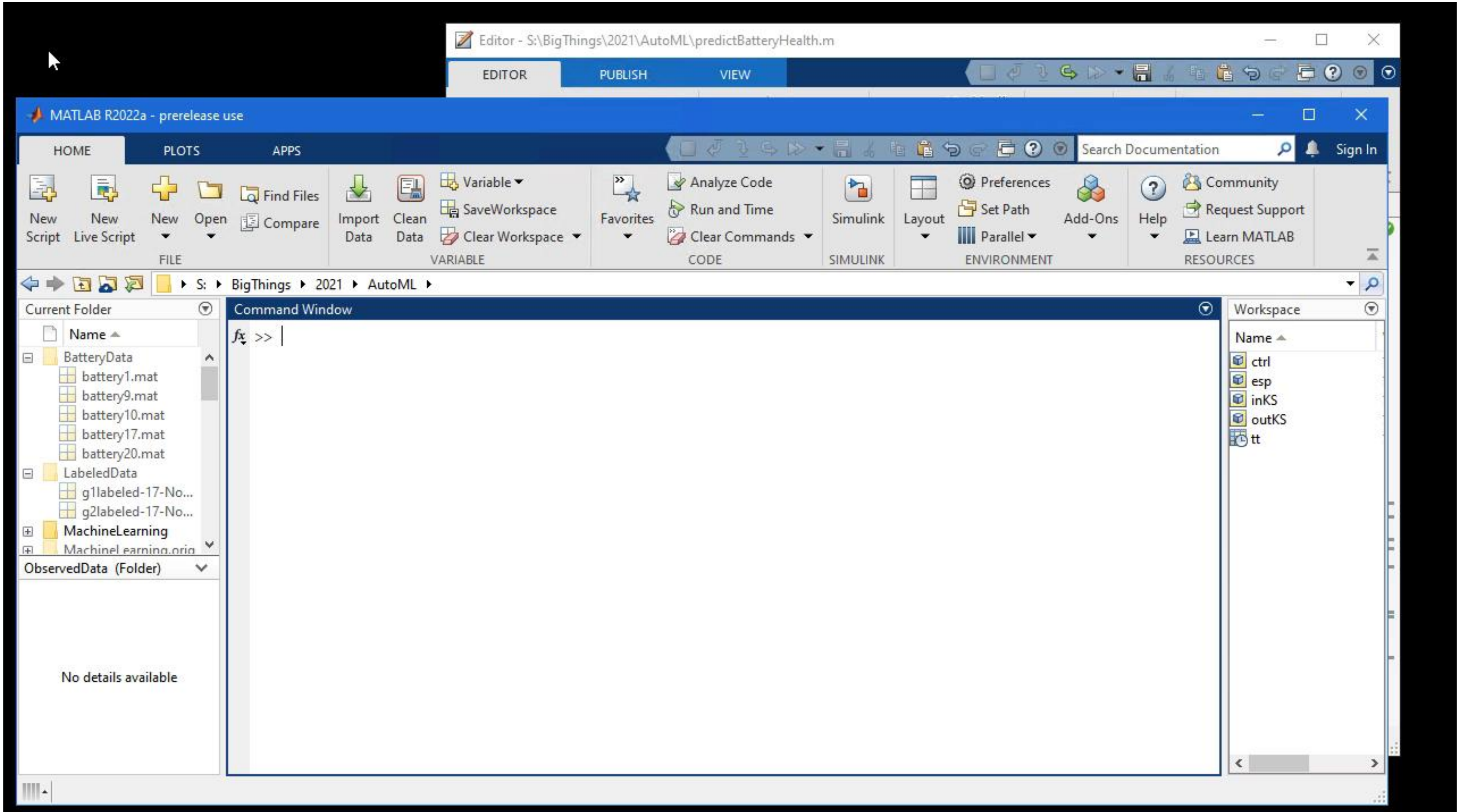
Per-battery stream partitioning enables horizontal scaling.



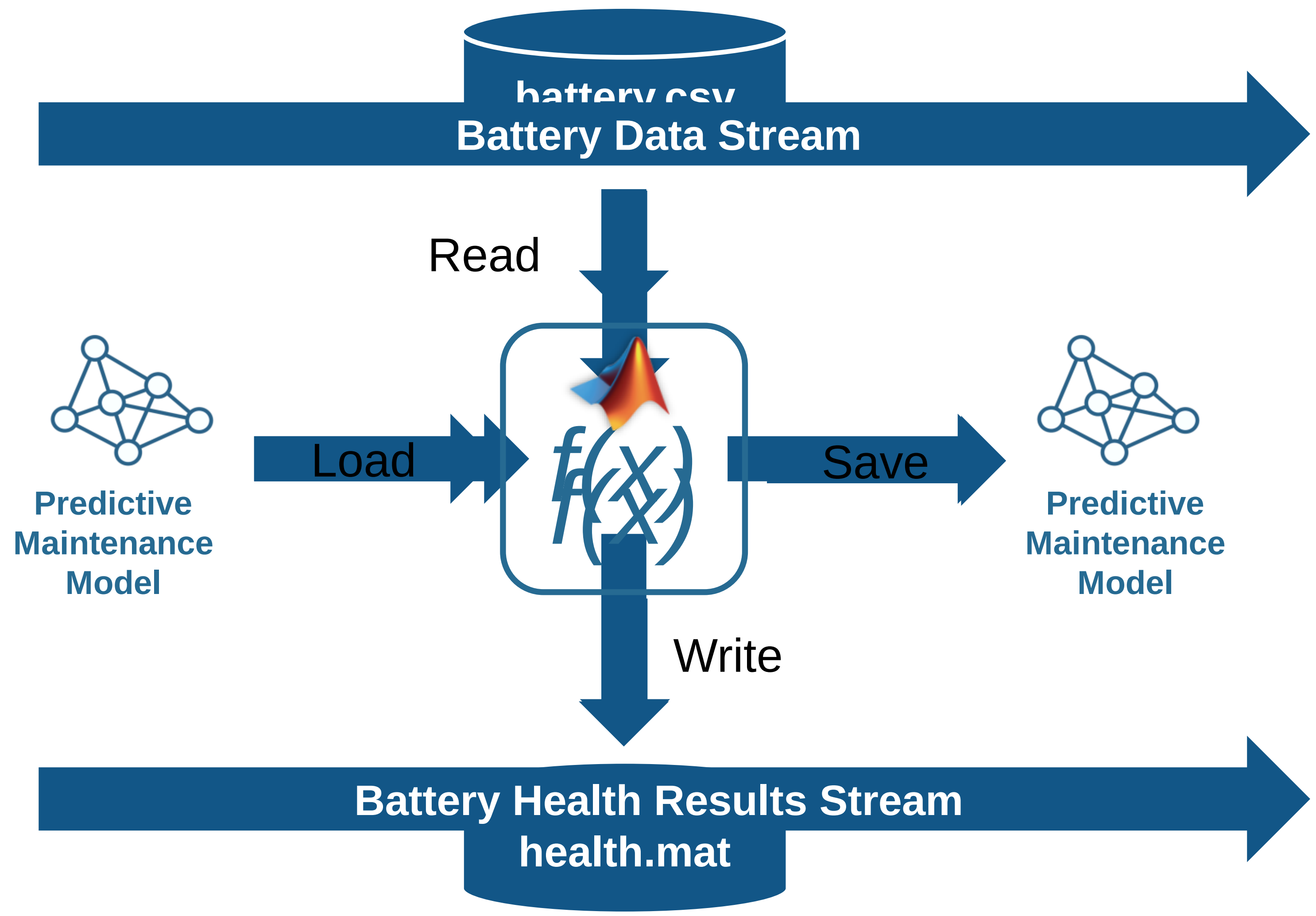
Model registry enables multiple predictors to update simultaneously.



Debug and test with desktop server before deploying to production.



Interactive access to streaming data simplifies model development.



Data and Results:
Streams
Files

Schema-controlled data import transforms JSON-encoded streaming data into native types.


```
"payload": {
    "Current": 0,
    "Voltage": 7.603372816643438,
    "T1": 294.728947368421,
    "T2": 294.728947368421,
    "SoC_B1": 0.5,
    "SoC_B2": 0.5,
    "SoH": 0,
    "key": 2
}
```

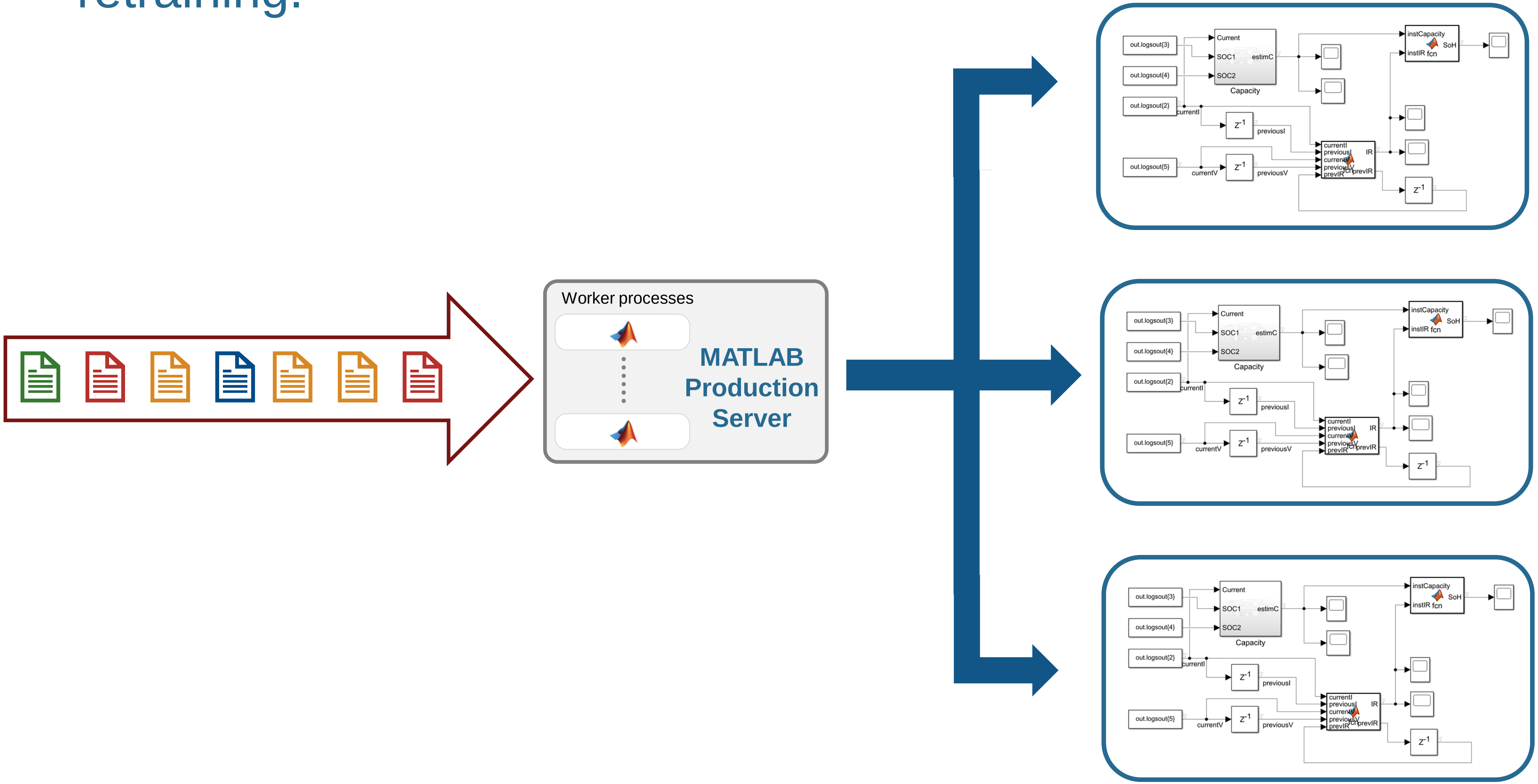
```
"schema": [
   {
      "name": "Current",
      "type": "double",
      "size": [
         1,
         1
      ],
      "missingValue": 0,
      "categorical": false
   },
```


```
>> ks = kafkaStream(host, port, topic);
>> tt = readtimetable(ks)
```

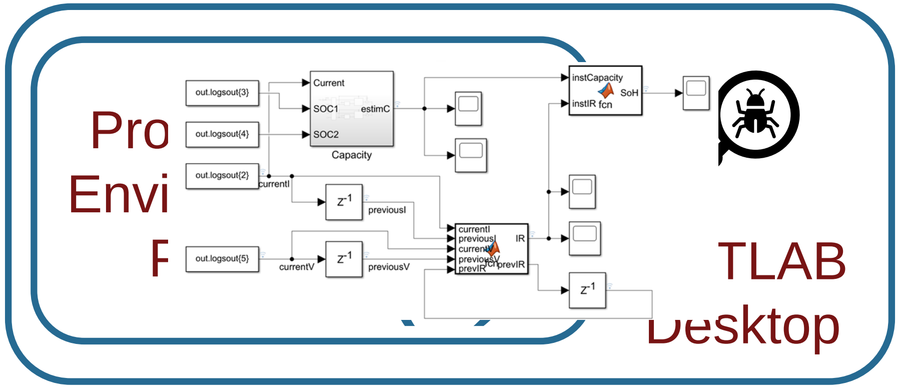
1800×8 timetable

timestamp	Current	Voltage	T1	T2
01-Nov-2021 00:00:00	0	7.6034	307.36	307.36
01-Nov-2021 00:00:01	2.6958	7.4188	307.36	307.48
01-Nov-2021 00:00:02	2.6961	7.4182	307.37	307.59
01-Nov-2021 00:00:03	2.6963	7.4175	307.38	307.69
01-Nov-2021 00:00:04	2.6966	7.4168	307.39	307.79
:	:	:	:	:
01-Nov-2021 00:29:55	-2.534	7.8926	309.64	311.09
01-Nov-2021 00:29:56	-2.5339	7.8929	309.64	311.09
01-Nov-2021 00:29:57	-2.5338	7.8932	309.64	311.09
01-Nov-2021 00:29:58	-2.5337	7.8935	309.64	311.09
01-Nov-2021 00:29:59	-2.5336	7.8939	309.64	311.09

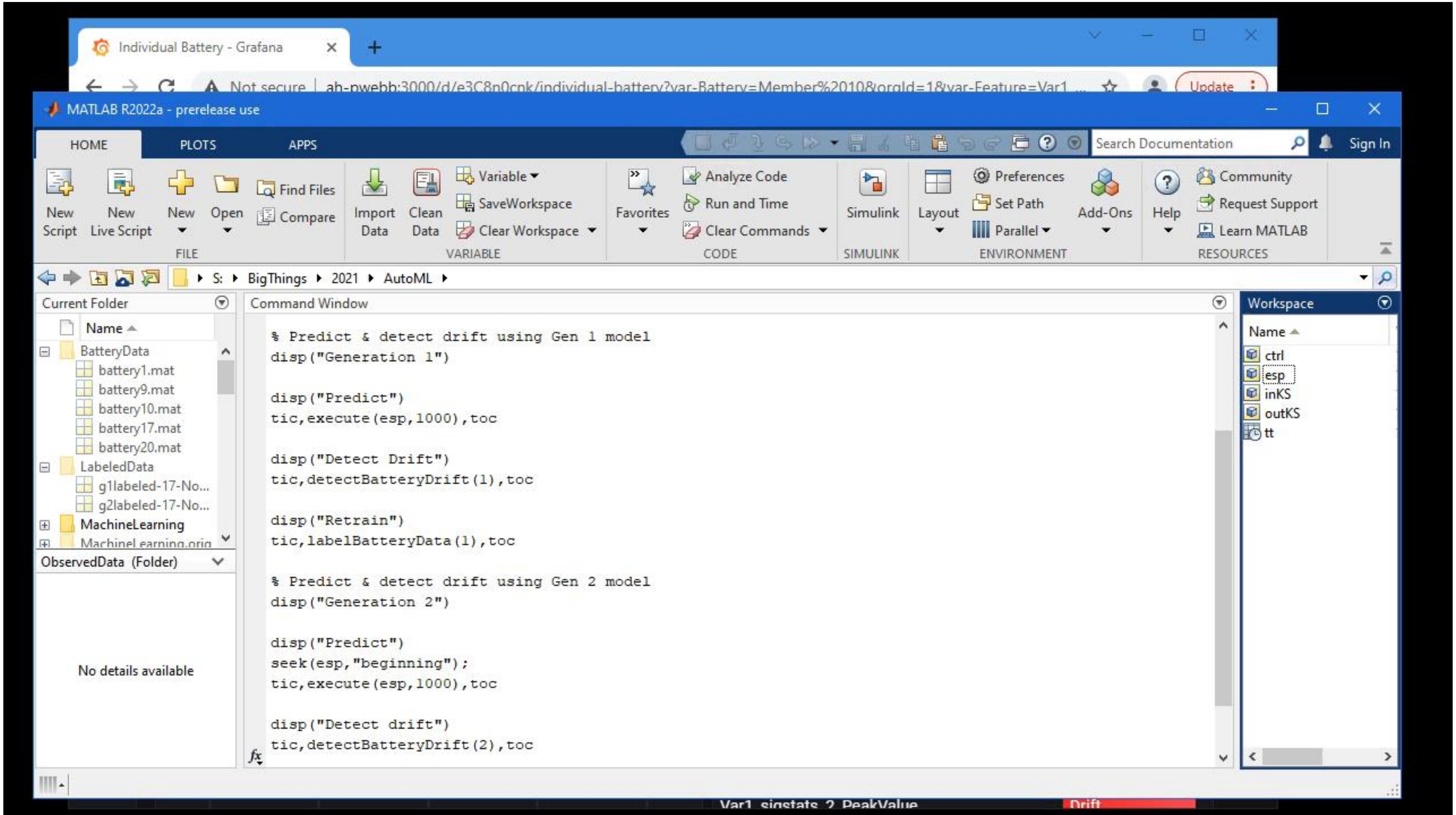
Deployable physical models enable automation and speed retraining.



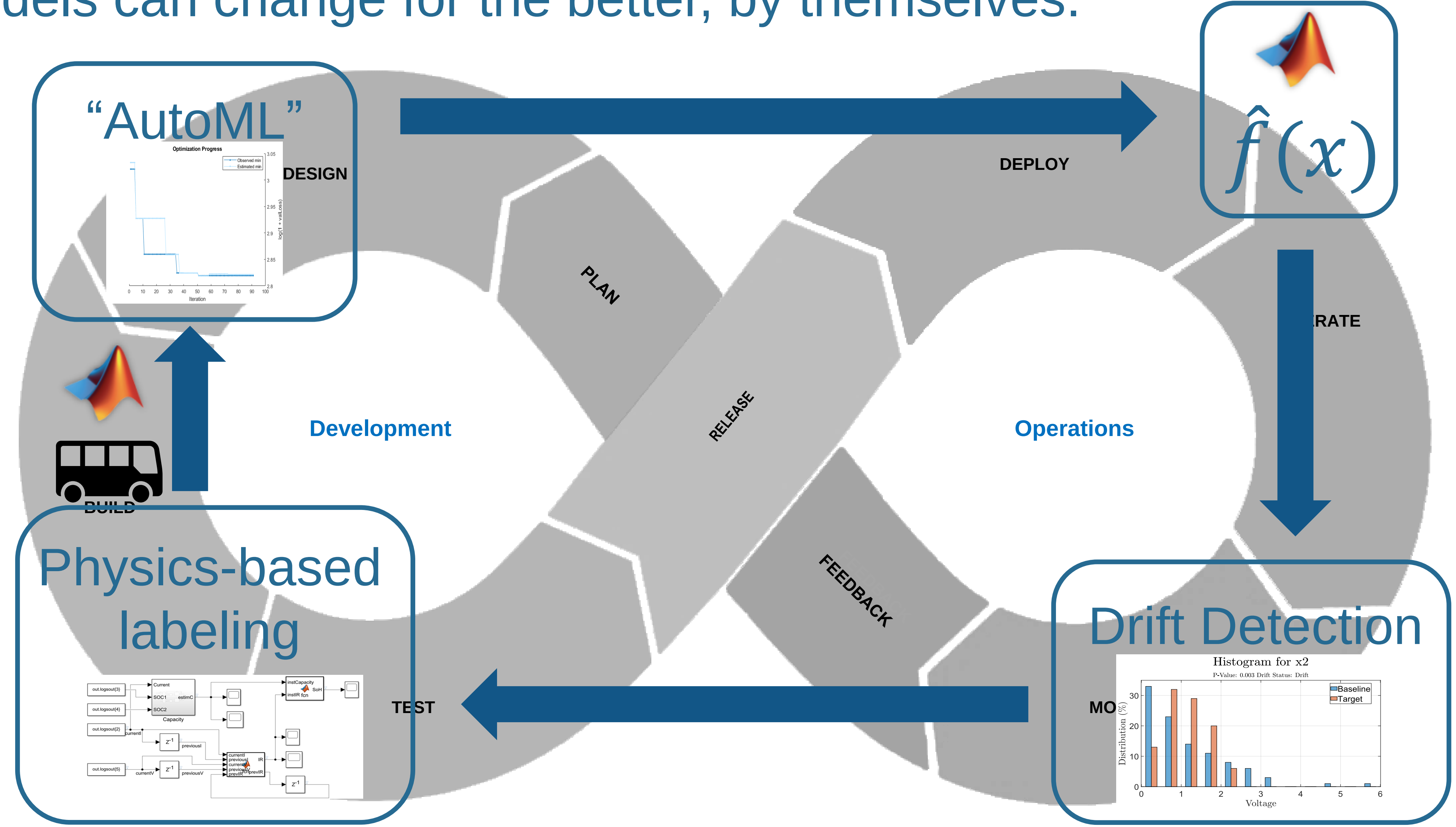
Your development toolchain needs a virtual production environment, native access to streams and deployable physical models.



Automating the Dev Ops cycle



Automate the entire Dev Ops cycle and your machine learning models can change for the better, by themselves.



MATLAB EXPO

Thank you

Dr Rishu Gupta, MathWorks

Peeyush Pankaj, MathWorks

MathWorks