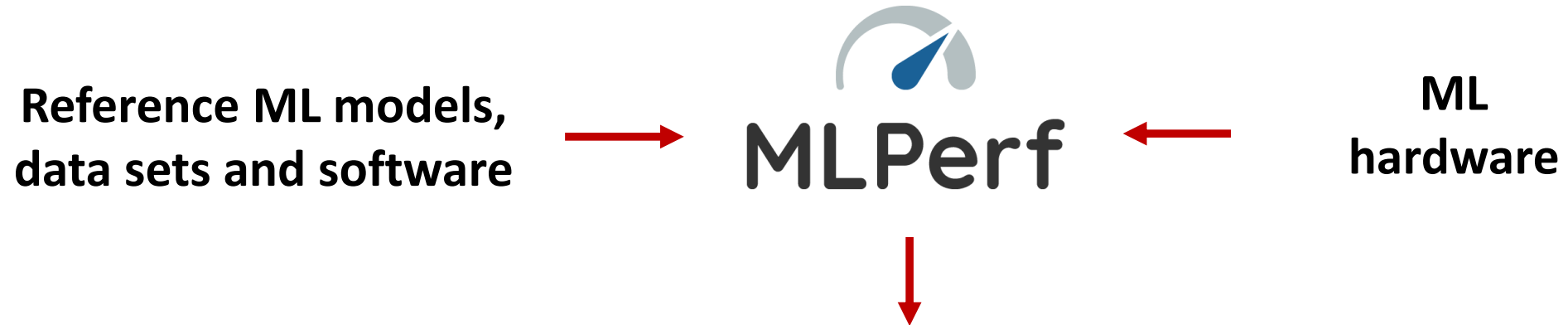


Preparing a new MLCommons workgroup to make it easier to run, customize and reproduce MLPerf benchmarks:

bit.ly/mlperf-edu-wg



Help users co-design and deploy efficient ML Systems in the real world

Agenda

github.com/mlcommons/ck/issues/325

- 1) Motivation for this WG and our automation meta-framework
 - Our experience reproducing ML papers and MLPerf benchmarks in the real world
 - Our motivation for a **non-intrusive** automation: Collective Knowledge concept (CK) – automated > 90% of MLPerf inference benchmark open division submissions
 - Intro to the latest MLCommons Collective Mind automation meta-framework (CM):
<https://github.com/mlcommons/ck>
- 2) Our on-going developments to modularize MLPerf, make it easier to plug any ML model, data set, framework, SDK into loadgen, and help you automate submissions, reproduce results and deploy efficient ML Systems
- 3) Open discussion (today, during next meetings and off-line):
 - How this WG can help you and the community
 - How you can help this WG

Pareto-efficient ML Systems in the real world

Our experience preparing and running MLPerf since the beginning

How to download and install correct versions of all components? How to connect all pieces together?
How to plug new components? How to run iterative experiments, reproduce them and submit best results?

Open-source MLCommons benchmarking infrastructure

Reference ML tasks

Vision Speech Recommendation Language

Reference ML model

ResNet50 3D UNET RNNT DLRM BERT

Reference data set

ImageNet KITS LibriSpeech Criteo Terabyte squad

ML engines, hardware-specific SDKs, allowed optimizers

PyTorch TF ONNX TVM QAIC CUDA OpenVINO

MLPerf benchmarking implementation with loadgen

Offline Server SingleStream MultiStream

Run MLPerf benchmarks on diverse platforms

Inference (cloud/edge) Mobile TinyML Training

Scripts/READMEs to pull Git repo and dependencies

Scripts/READMEs to download and process models

Scripts/READMEs to download and process datasets

Scripts/READMEs to install correct SW versions

Scripts/READMEs to prepare CMD to run benchmark

Scripts/READMEs to process outputs, validate results and prepare submissions

Learnings from reproducing ML and Systems papers since 2014: cTuning.org/ae

ACM TechTalk'21 "Reproducing 150 Research Papers and Testing Them in the Real World: Challenges and Solutions"

Grigori Fursin and Peter Mattson (<https://youtu.be/7zpeIVwICa4>)

Open-source MLCommons benchmarking infrastructure

Reference ML tasks

Vision Speech Recommendation Language

Reference ML model

ResNet50 3D UNET RNNT DLRM BERT

Reference data set

ImageNet KITS LibriSpeech Criteo Terabyte squad

ML engines, hardware-specific SDKs, allowed optimizers

PyTorch TF ONNX TVM QAIC CUDA OpenVINO

MLPerf benchmarking implementation with loadgen

Offline Server SingleStream MultiStream

Run MLPerf benchmarks on diverse platforms

Inference (cloud/edge) Mobile TinyML Training

Containers: very useful as a snapshot but can't adapt to evolving SW/HW

Scripts/READMEs to pull Git repo and dependencies

Scripts/READMEs to download and process models

Scripts/READMEs to download and process datasets

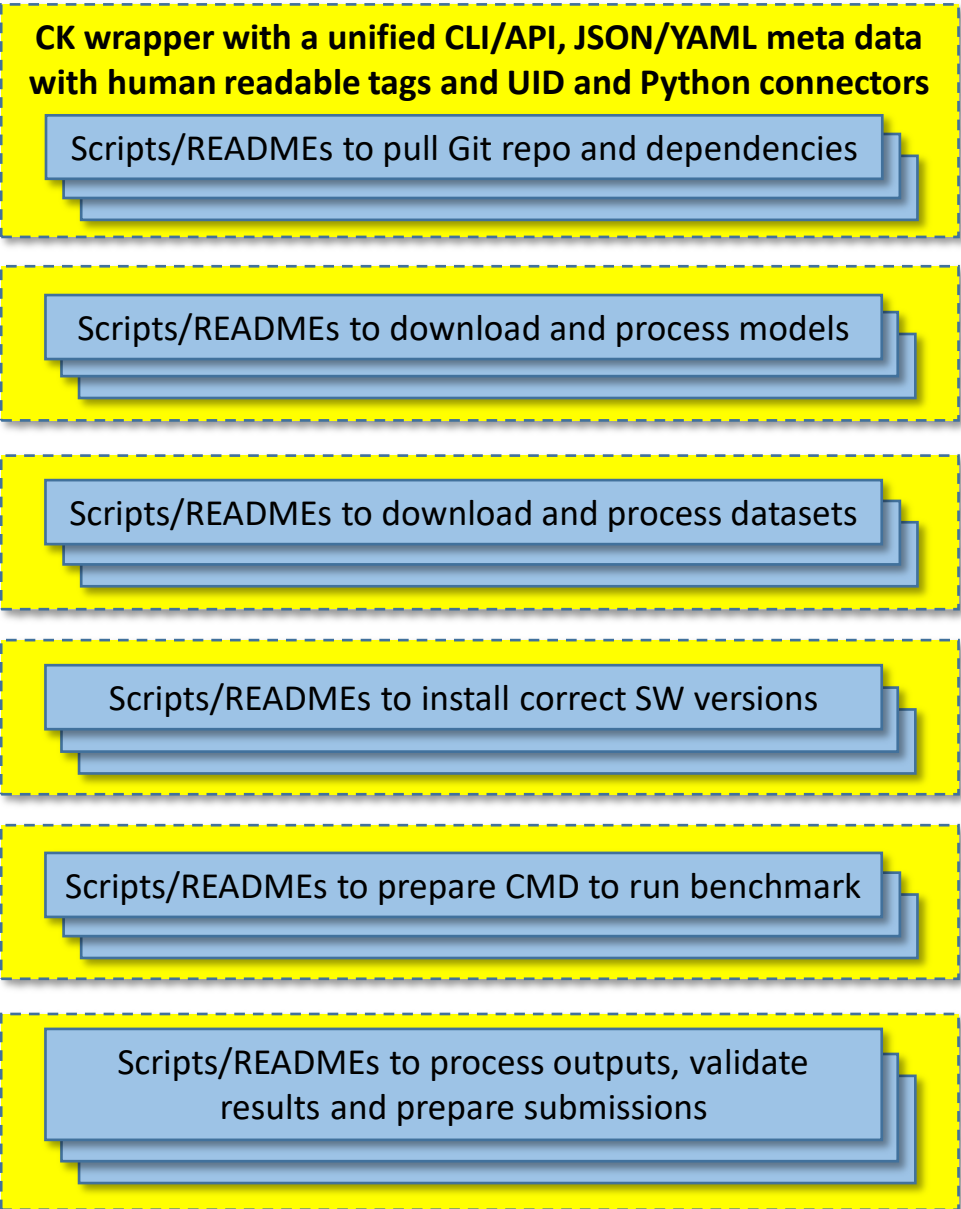
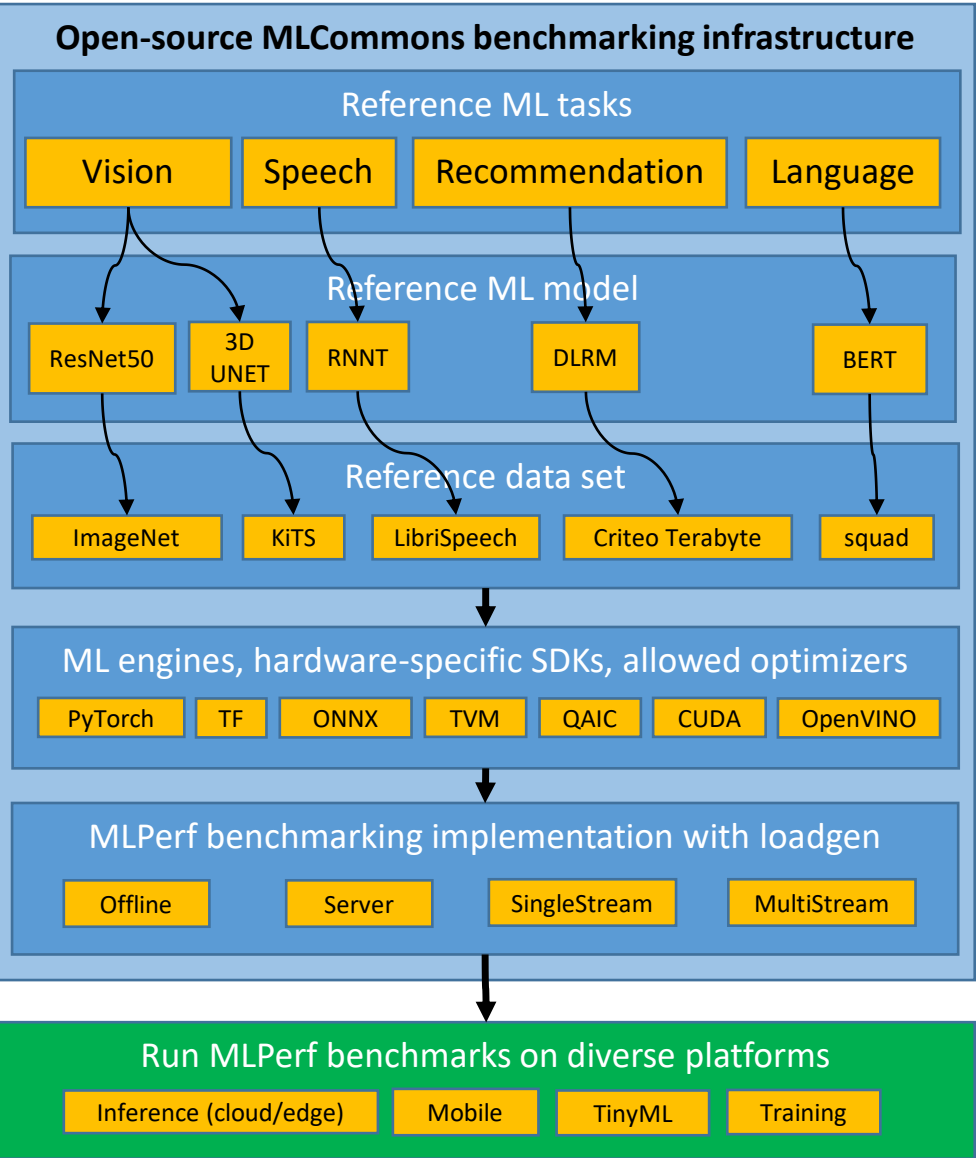
Scripts/READMEs to install correct SW versions

Scripts/READMEs to prepare CMD to run benchmark

Scripts/READMEs to process outputs, validate results and prepare submissions

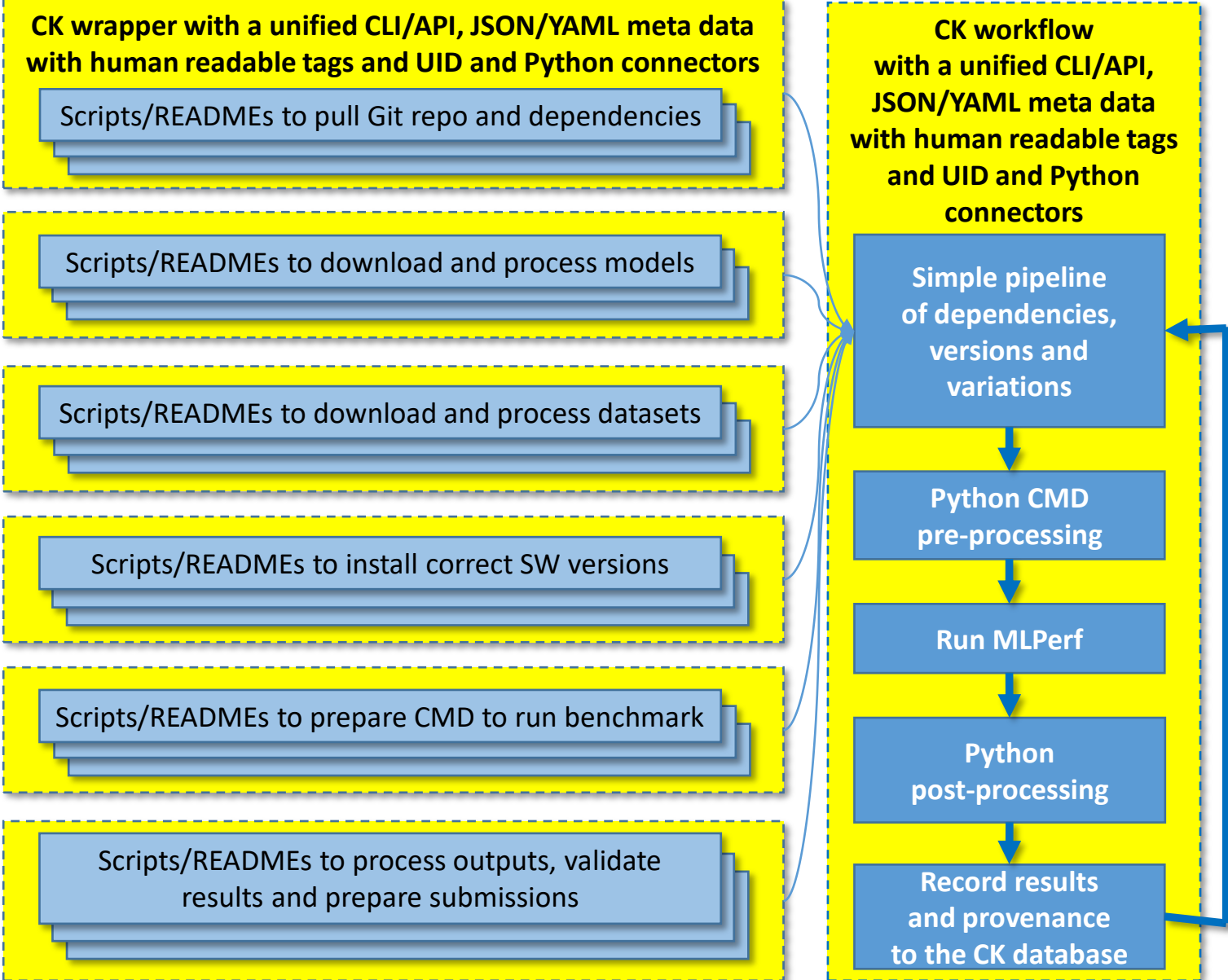
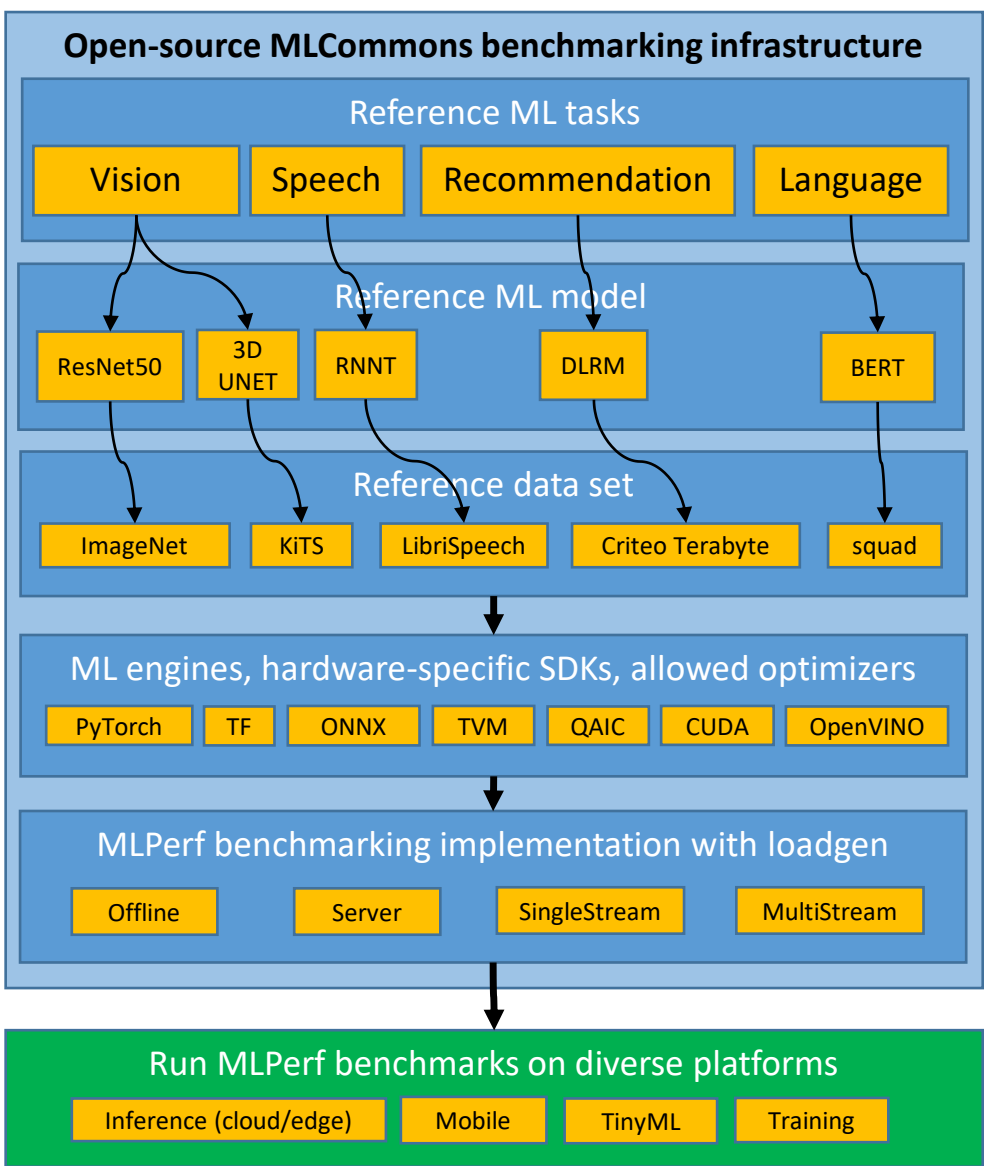
Collective Knowledge concept - non-intrusive and reusable automation: arxiv.org/abs/2011.01149

Turn projects, artifacts and scripts into a database of portable and reusable components with minimal or zero effort:



Automate MLPerf steps and make them more reusable and deterministic

CK prototype helped to automate > 90% of MLPerf inference open division submissions and some closed ones (Qualcomm, Krai, Dell, HPE, Lenovo and OctoML)



Donated Collective Knowledge to MLCommons to benefit everyone and continue collaborative developments

Grigori Fursin

OctoML, cTuning foundation

grigori@octoml.ai

Collaborated with ACM
to reproduce 150+ ML and
Systems papers
and validate them
in the real world.

Now developing CM to help
companies reproduce MLPerf
results and deploy efficient ML
Systems in production.

Arjun Suresh

OctoML

asuresh@octoml.ai

Used CK at Krai to automate and
optimize Qualcomm AI 100
MLPerf inference submissions.
Joined OctoML to develop CM
as a community effort.

Thomas Zhu

Oxford University

thomas.zhu.sh@hotmail.com

Worked with Vijay Janapa Reddi
at Harvard to reproduce MLPerf
inference benchmarks.

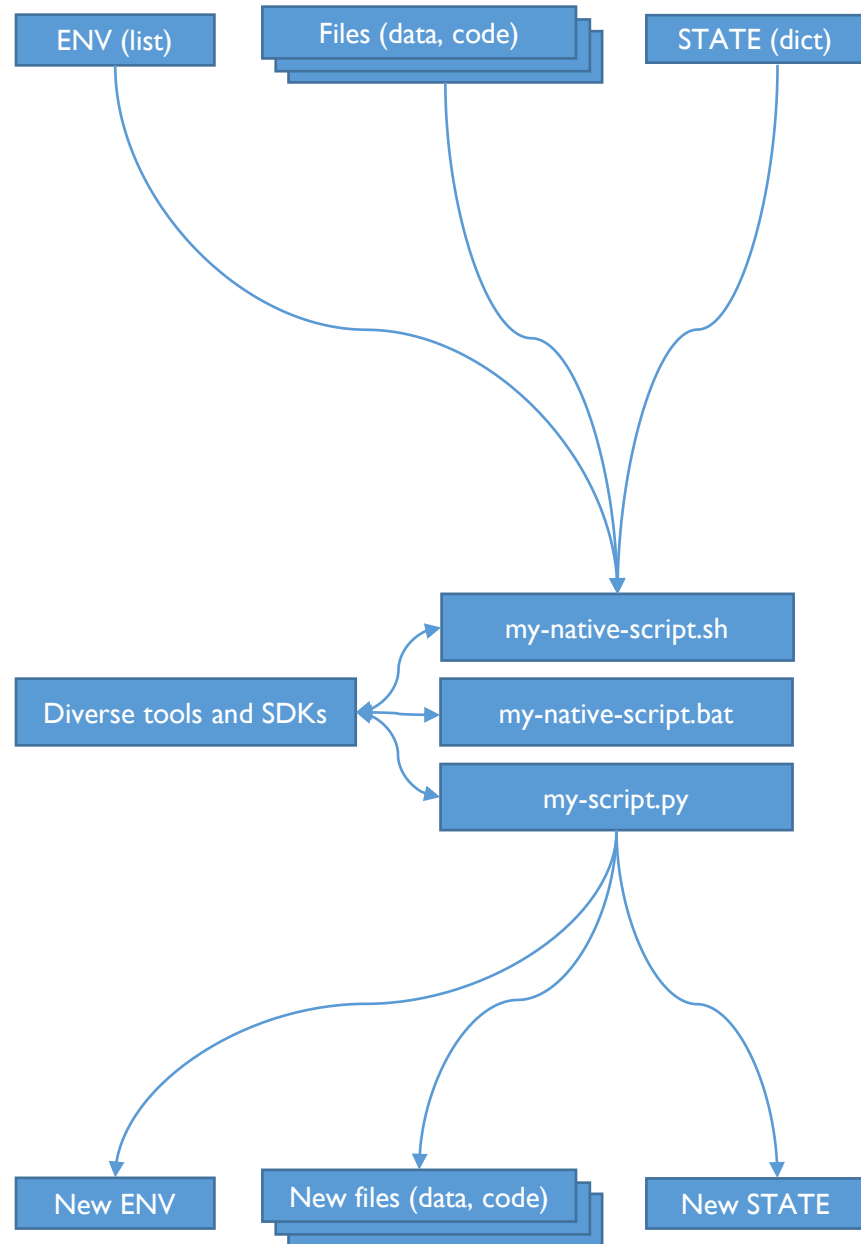
Joined collaborative CM effort to
make it easier to run and
reproduce MLPerf benchmarks.

Now developing universal
plug&play C++ front-end for
loadgen powered by CM...

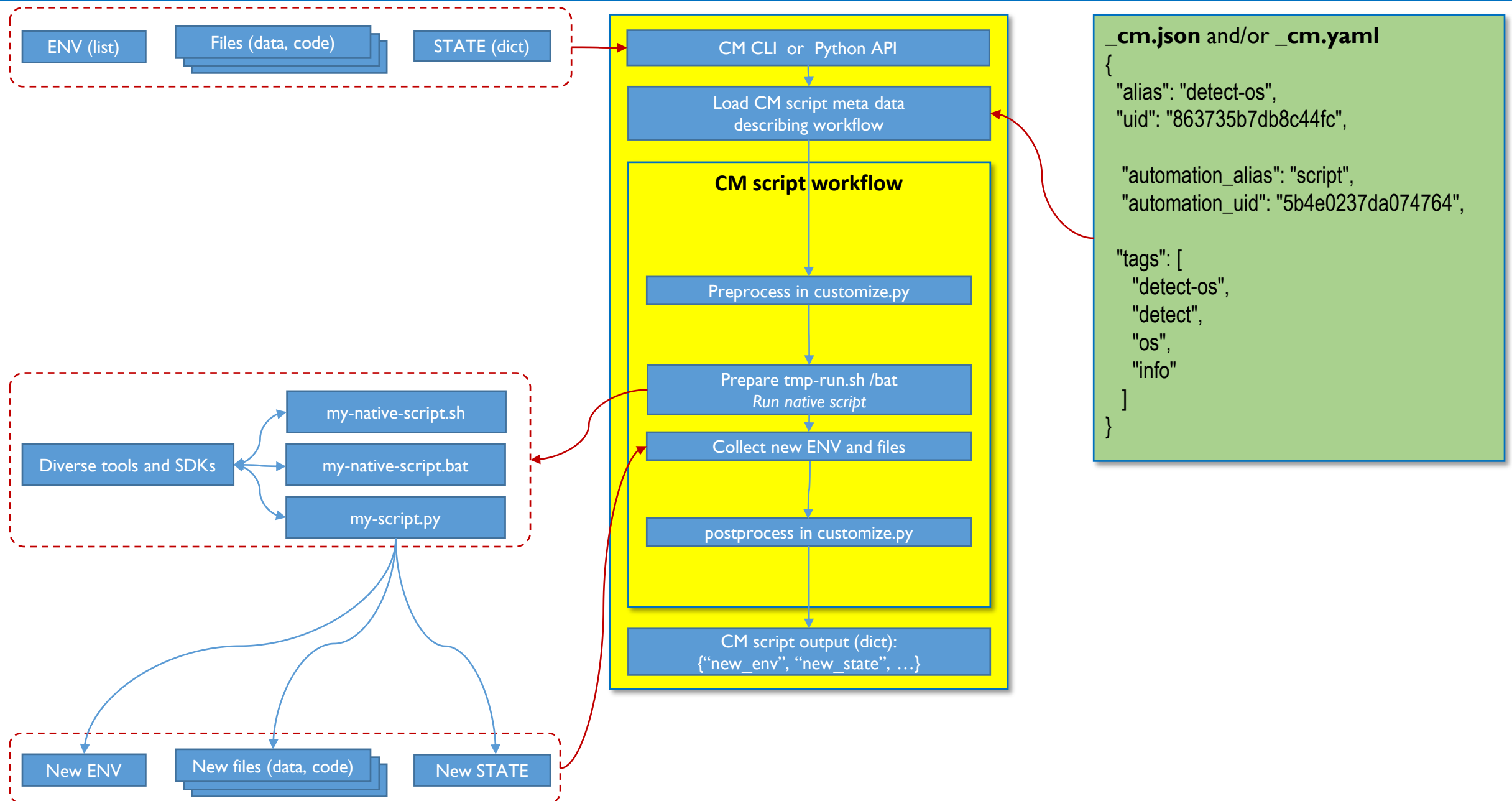
We joined forces to develop the **MLCommons Collective Mind automation meta-framework (CM)**:
a simplified implementation of the CK concept to make it easier to understand, reuse and extend!

Stable version released on September 14, 2022: <https://github.com/mlcommons/ck>

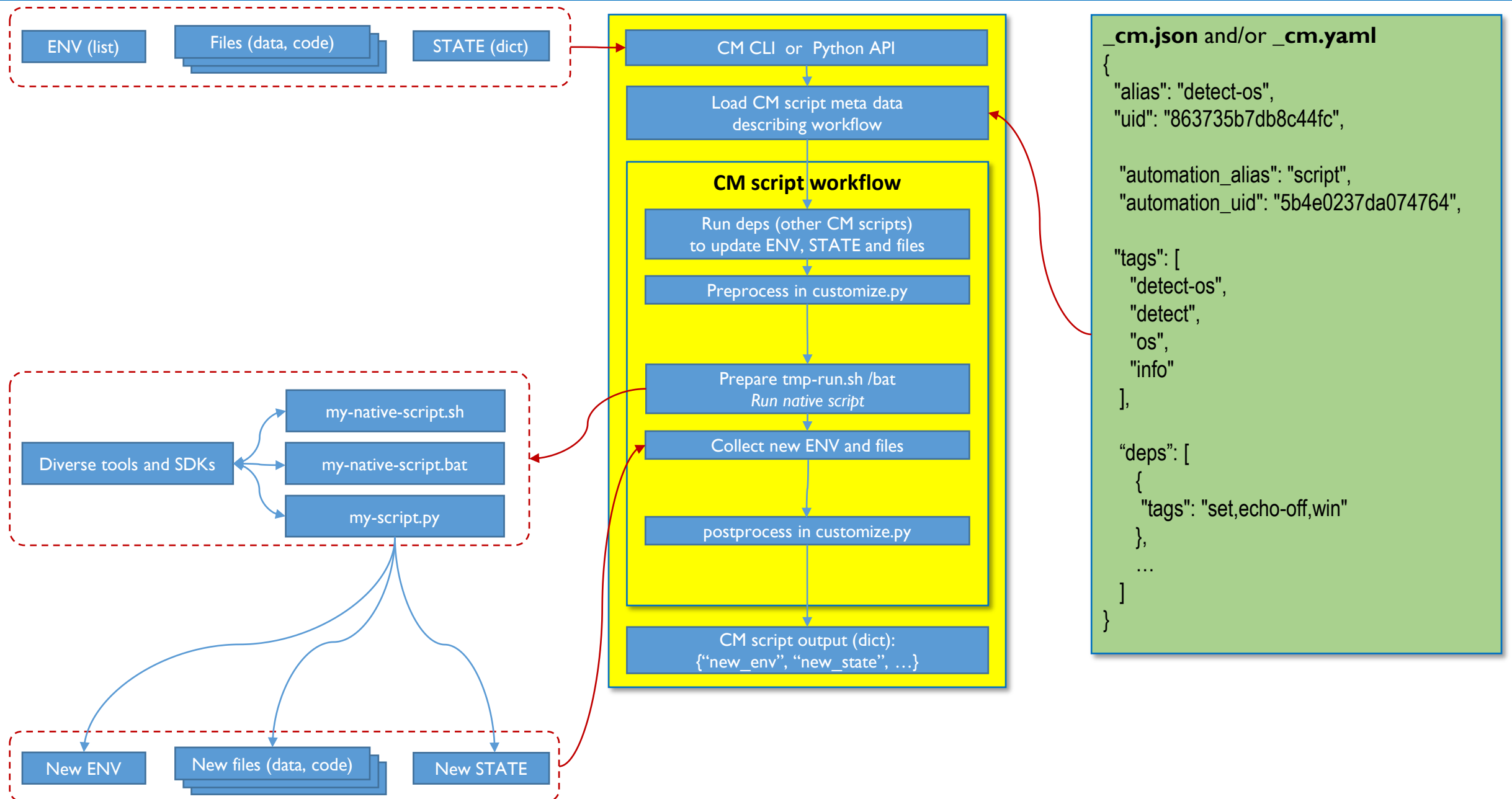
CM goal: make MLPerf scripts and artifacts more reusable, portable, customizable and deterministic



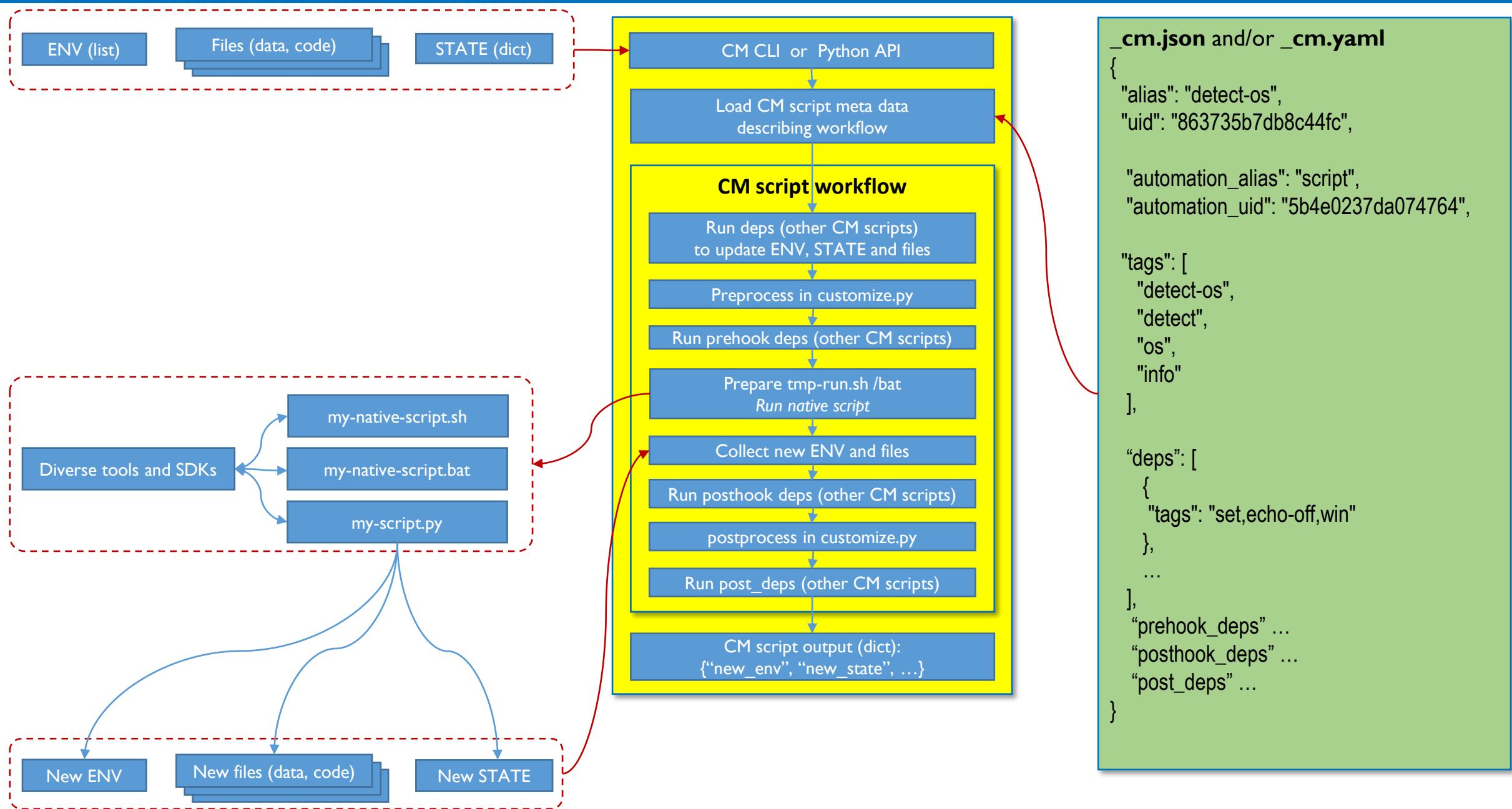
CM non-intrusive “LEGO bricks”: making scripts and artifacts more reusable, portable, customizable and deterministic



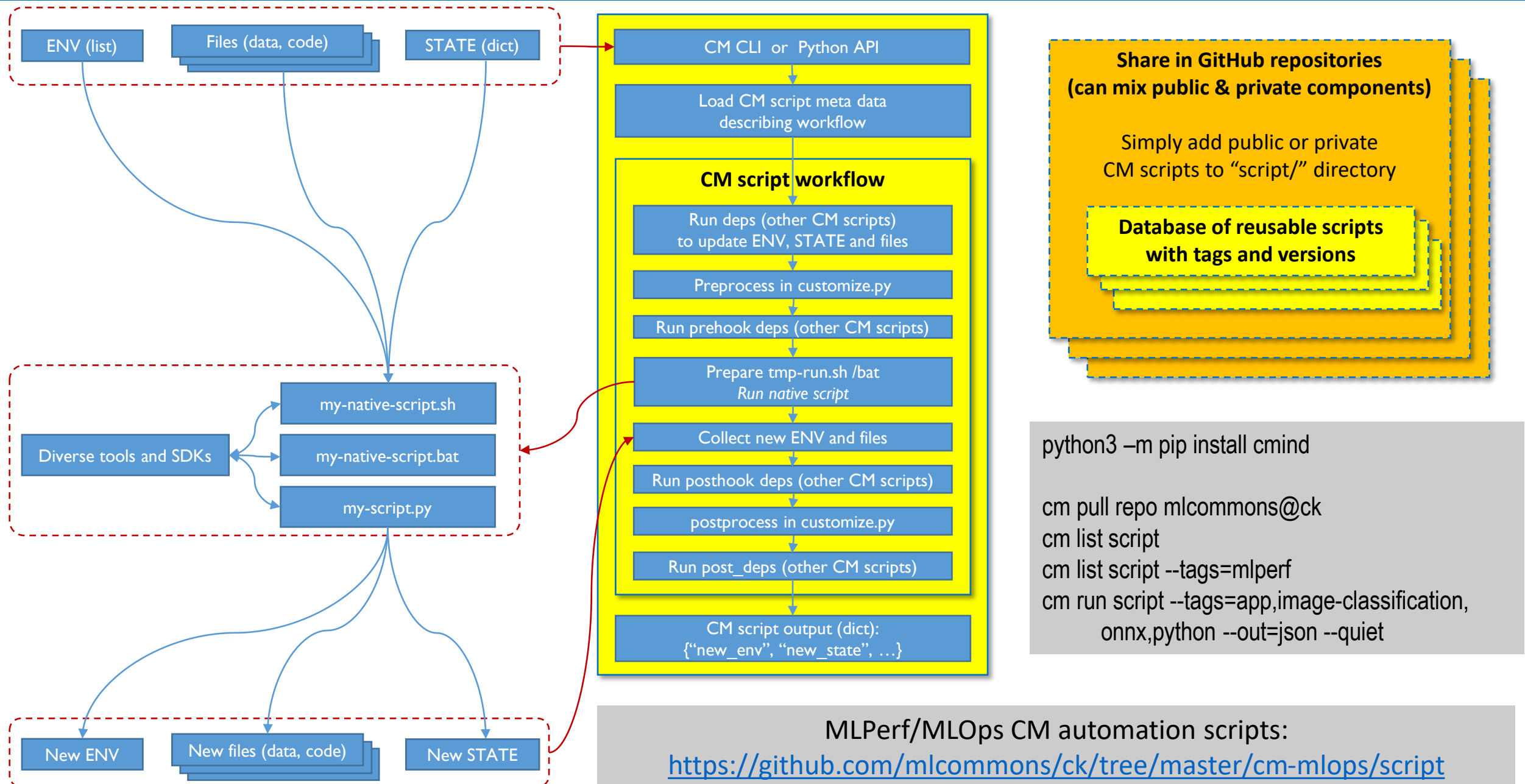
CM non-intrusive “LEGO bricks”: making scripts and artifacts more reusable, portable, customizable and deterministic



CM non-intrusive “LEGO bricks”: making scripts and artifacts more reusable, portable, customizable and deterministic



CM non-intrusive “LEGO bricks”: making scripts and artifacts more reusable, portable, customizable and deterministic



Develop a hierarchy of reusable, portable and customizable CM scripts to adapt ML Systems to any user environment

We continue developing and sharing all CM scripts at <https://github.com/mlcommons/ck/tree/master/cm-mlops/script> .

Simple image classification example that can run natively on any user platform
with Linux (Ubuntu, Red Hat, Debian), Windows and MacOS
while automatically adapting to a given software, hardware and data:

<https://github.com/mlcommons/ck/tree/master/cm-mlops/script/app-image-classification-onnx-py>

```
python3 -m pip install cmind
```

```
cm pull repo mlcommons@ck
```

```
cm run script --tags=app,image-classification,onnx,python (--input=my-image.jpg)
```

```
cm show cache --tags=ml-model
```

```
cm run script --tags=detect,os --out=json
```

```
cm run script --tags=get,python --version=3.10
```

```
cm run script --tags=install,llvm,prebuilt --version_min=13.0.0 --version_max=14.0.0
```

```
cm run script --tags=get,ml-model-onnx,resnet50
```

```
cm run script --tags=get,dataset,imagenet,original,_2012-500
```

Tutorials: <https://github.com/mlcommons/ck#tutorials>

Generate modular and adaptive containers with CM scripts and CM APIs (collaboration with MLCube)

We continue developing and sharing all CM scripts at <https://github.com/mlcommons/ck/tree/master/cm-mlops/script> .

Simple image classification example that can run natively on any user platform
with Linux (Ubuntu, Red Hat, Debian), Windows and MacOS
while automatically adapting to a given software, hardware and data:

<https://github.com/mlcommons/ck/tree/master/cm-mlops/script/app-image-classification-onnx-py>

```
python3 -m pip install cmind
```

```
cm pull repo mlcommons@ck
```

```
cm run script --tags=app,image-classification,onnx,python (--input=my-image.jpg)
```

```
cm show cache --tags=ml-model
```

```
cm run script --tags=detect,os --out=json
```

```
cm run script --tags=get,python --version=3.10
```

```
cm run script --tags=install,llvm,prebuilt --version_min=13.0.0 --version_max=14.0.0
```

```
cm run script --tags=get,ml-model-onnx,resnet50
```

```
cm run script --tags=get,dataset,imagenet,original,_2012-500
```

tutorials: <https://github.com/mlcommons/ck#tutorials>

Why are we preparing this MLPerf education and reproducibility workgroup?

<https://github.com/mlcommons/ck/blob/master/docs/mlperf-education-workgroup.md>

<https://github.com/mlcommons/ck/issues/325>

We would like to share our experience and open-source CM playground to learn with the community how to

- develop an automated open-source workflow to make it easier to plug any real-world ML & AI tasks, models, data sets, software and hardware into the MLPerf benchmarking infrastructure;
- use this workflow to help the newcomers learn how to customize and run MLPerf benchmarks across rapidly evolving software, hardware and data;
- lower the barrier of entry for new MLPerf submitters and reduce their associated costs;
- automate design space exploration of diverse ML/SW/HW stacks to trade off performance, accuracy, energy, size and costs; automate submission of Pareto-efficient configurations to MLPerf;
- help end-users visualize all MLPerf results, reproduce them and deploy the most suitable ML/SW/HW stacks in production;
- support reproducibility initiatives and optimization/co-design competitions at ML and Systems conferences using rigorous MLPerf methodology and our educational toolkit.

Current moderators:

- [OctoML \(Grigori and Arjun\)](#)
- *Will be looking for a motivated person from another organization – please get in touch if you are interested!*

How can you help? We work with volunteers to

- add reusable CM scripts to download more ML models and unify/extend their meta description:

```
"env": {  
  "ML_MODEL_DATA_LAYOUT": "NCHW",  
  "ML_MODEL_GIVEN_CHANNEL_MEANS": "123.68 116.78 103.94",  
  "ML_MODEL_SUBTRACT_MEAN": True,  
  "ML_MODEL_NORMALIZE_DATA": 0,  
  "ML_MODEL_IMAGE_HEIGHT": 224,  
  "ML_MODEL_IMAGE_WIDTH": 224,  
  "ML_MODEL_INPUT_LAYER_NAME": "input_tensor:0",  
  "ML_MODEL_OUTPUT_LAYER_NAME": "softmax_tensor:0",  
  "ML_PACKAGE_NAME": "resnet50_v1_softmax.onnx.gz",  
  "ML_PACKAGE_URL": https://zenodo.org/...  
},  
"tags": ["ml-model", "image-classification", "onnx", "nchw", "resnet50", "side.224", "mlperf" ...]
```

- add reusable CM scripts to download and pre-process more data sets
- add reusable CM scripts to detect/install more ML framework, libraries, SDKs
- add more MLPerf tasks
- add support for GPUs, TPUs, mobile devices, edge devices, etc
- add quantization and NAS scripts
- add visualization of all MLPerf results and automatic synthesis of Pareto-efficient ML Systems

Meeting notes: [Google docs](#)

MLPerf inference benchmark automation and reproducibility workflow

- *Reference Python implementation image classification*

On-going: [[GitHub ticket](#)] [[GitHub ticket2](#)] [[Reproducibility matrix](#)]

Current status: Image classification (ResNet50) is working with Tensorflow and Onnxruntime, Object Detection (Retinanet) is working with Onnxruntime and Pytorch

- *C++ loadgen plug&play implementation for any MLPerf benchmark*

On-going internship: [[GitHub ticket](#)]

Current status: Image classification (ResNet50) is working with onnxruntime

MLPerf inference benchmark submission workflow

[Demo of the CM script to automate OctoML's v2.1 submission](#) - will be reused and improved to generalize submissions.

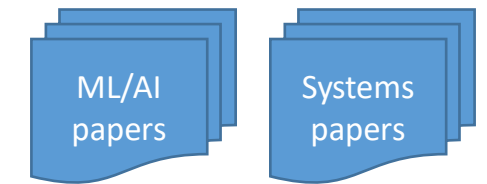
For image classification and object detection we have added automation for the full stack of operations - getting SUT details, getting all the valid scenarios, getting the scenario configurations, running the scenario, calling the accuracy script and accuracy truncation script, doing the required compliance runs and finally generating the submission tree.

Meeting notes: [Google docs](#)

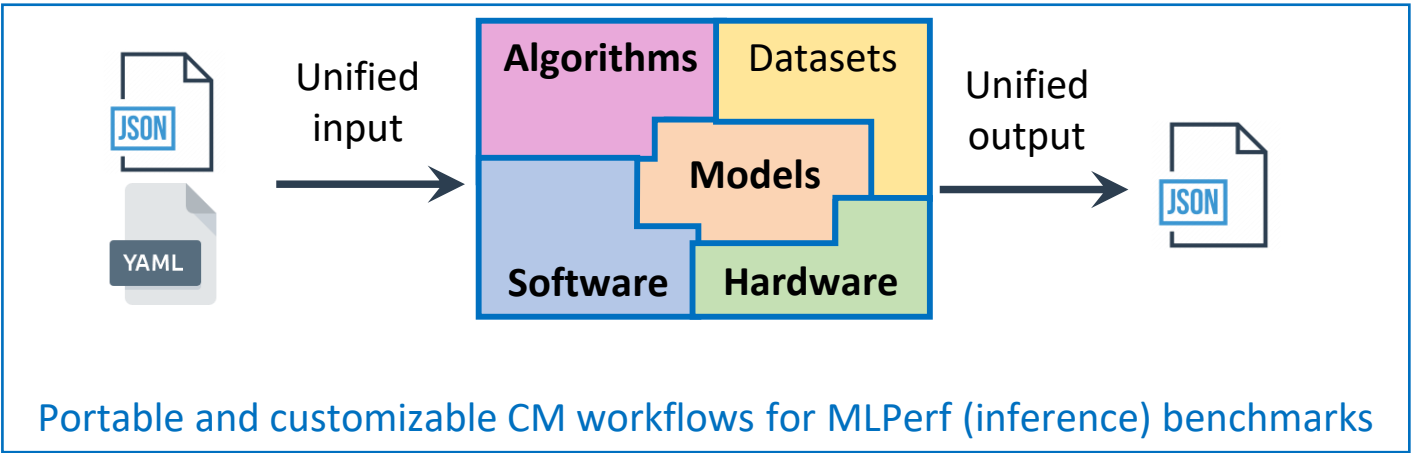
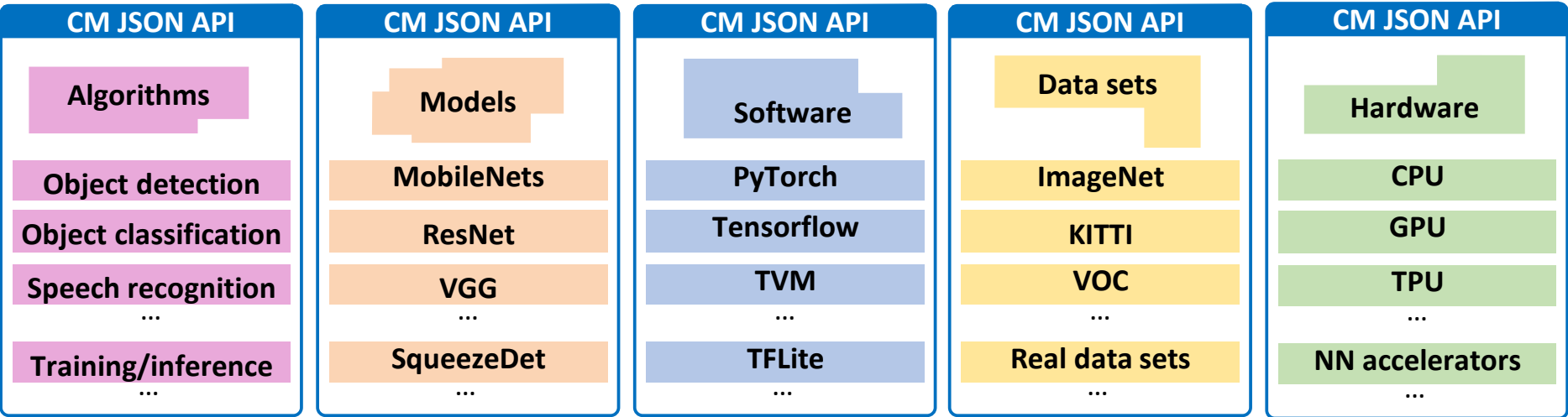
- **Add power measurements to the above CM-based workflows**
Waiting for Power meter Yokogawa WT310e to be integrated with CM MLPerf workflows
- Create MLPerf automation tutorials for the newcomers
- MLPerf design space exploration workflow (automate open division submissions)
- MLPerf visualization of all past and new results with Pareto frontier [[See example](#)]
- Cross-workgroup collaborations reusing common CM scripts and workflows:
 - Providing open implementation and CM workflows for “loadgen over network”
 - Generation of modular ML containers (MLCube) with CM components (Best Practices & MedPerf)
 - MLPerf TinyMLPerf benchmarking submission workflow
 - Automatic synthesis of the MLPerf Mobile app [[GitHub ticket](#)]
 - MLPerf open division competitions (Research WG)

On-going community developments: enable design space exploration of ML Systems using loadgen and plug&play CM

<https://github.com/mlcommons/ck/blob/master/docs/mlperf-automation-cm/reproducibility.md>



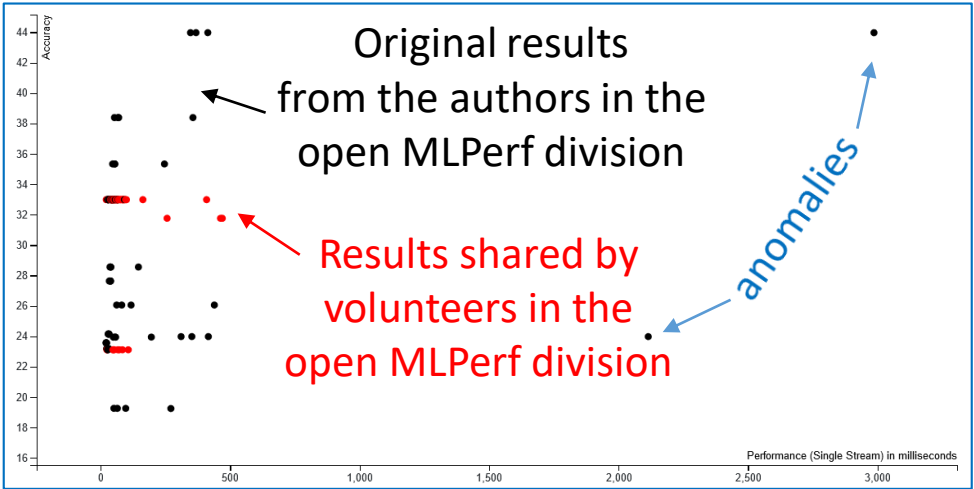
Share portable workflows, adaptive containers, automation actions and plug&play components along with research papers:



Quickly prototype and test ideas on any tech. stack



Help end-users reproduce MLPerf results and deploy Pareto-efficient ML/SW/HW stacks in the real world!



Open discussion

- How this workgroup can help you and your colleagues?
 - Web page for this group: <https://github.com/mlcommons/ck/blob/master/docs/mlperf-education-workgroup.md>
 - Meeting notes: https://docs.google.com/document/d/1zMNK1m_LhWm6jimZK6YE05hu4VH9usdbKJ3nBy-ZPAw
- How this workgroup can help you and your colleagues?
- How CM can automate all your repetitive MLPerf tasks and experiments and help you focus on innovation and optimization?
- How to communicate?
 - GitHub issues: <https://github.com/mlcommons/ck/issues>
 - Weekly conf-calls
- Contacts:
 - Don't hesitate to contact grigori@octoml.ai and asuresh@octoml.ai if you have any feedback – the goal of this community effort is to help everyone with the MLPerf automation, optimization and submission!