DLSpec: A Deep Learning Task Exchange Specification

Abdul Dakkak[1*], Cheng Li[1*], Jinjun Xiong[2], Wen-mei Hwu[1]

University of Illinois Urbana-Champaign[1], IBM Research[2]

OpML'2020

Background

- Deep Learning (DL) innovations are introduced at a fast pace
- Current lack of standard specification of DL tasks makes sharing, running, reproducing, and comparing DL innovations difficult

Current Practice of Publishing DL Artifacts

- Ad-hoc scripts and textual documentation to describe the execution process of *DL tasks*
  - Curation of DL tasks in framework model zoo
  - Model catalogs that can be used through a cloud provider's API
- Hard to reproduce the reported accuracy or performance results and have a consistent comparison across DL artifacts

DLSpec Objectives

- A DL artifact exchange specification with clearly defined model, data, software, and hardware aspects
  - Model-, dataset-, software-, and hardware agnostic
  - Works with runtimes built using existing MLOp tools
- We developed a DLSpec runtime for DL inference tasks in the context of benchmarking

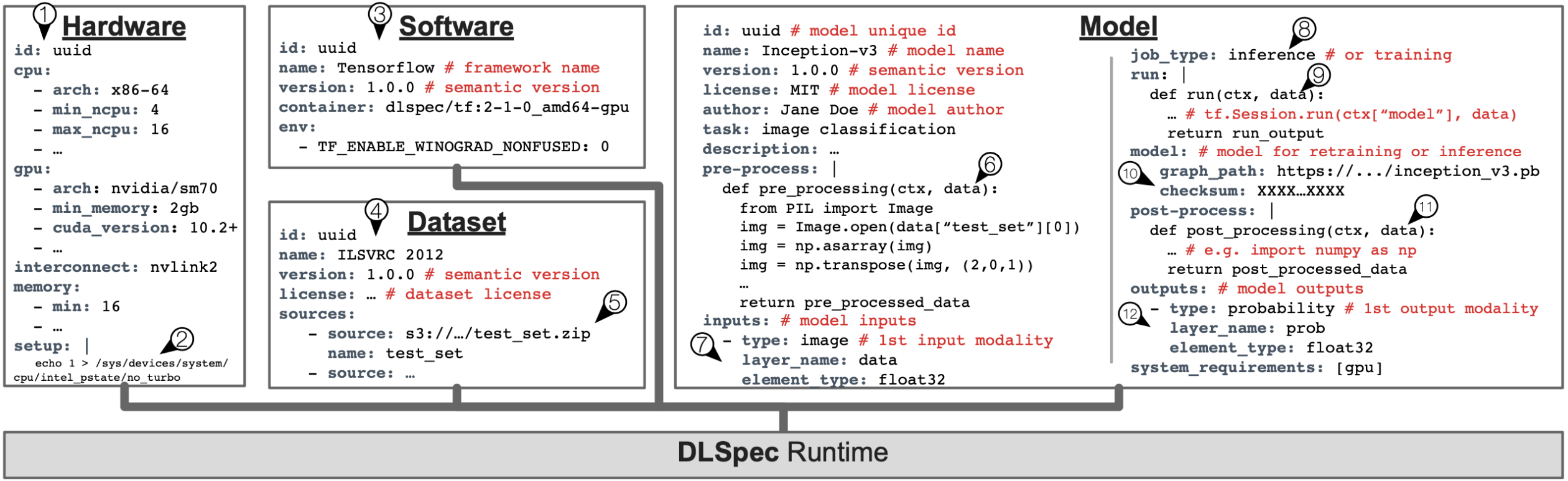
DLSpec is Based on a Few Key Principles

- Reproducible
- Minimal
  - Only contains essential information to increase the transparency and ease the creation
- Program-/human-readable
  - Executed by a runtime/easy to introspect and repurpose
- Maximum expressiveness
  - Describes both training and inference

DLSpec is Based on a Few Key Principles

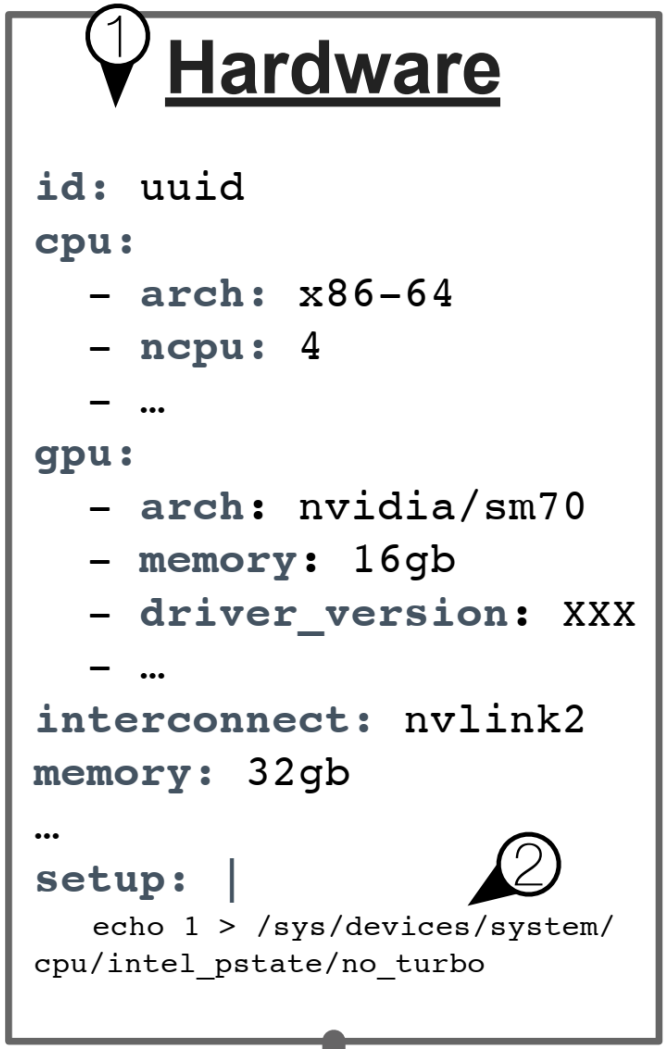
- Decoupling DL task description
  - Increases the reuse/portability and enables easy of comparison
- Splitting the DL task pipeline stages
  - Enables consistent comparison and simplifies accuracy and performance debugging
- Avoiding serializing intermediate data into files
  - Avoids high serializing/deserializing overhead
  - Supports DL tasks that use streaming data

DLSpec Design



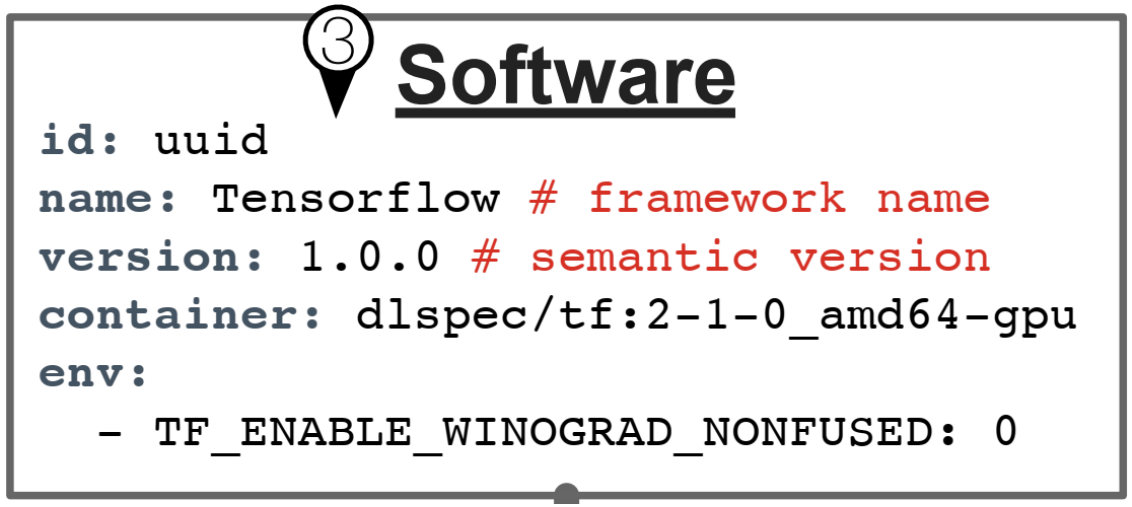
Hardware Manifest

- Defines the hardware requirements for a DL task
- Some hardware settings cannot be specified within a container (E.g. the runtime set Intel's turbo-boosting outside the container)



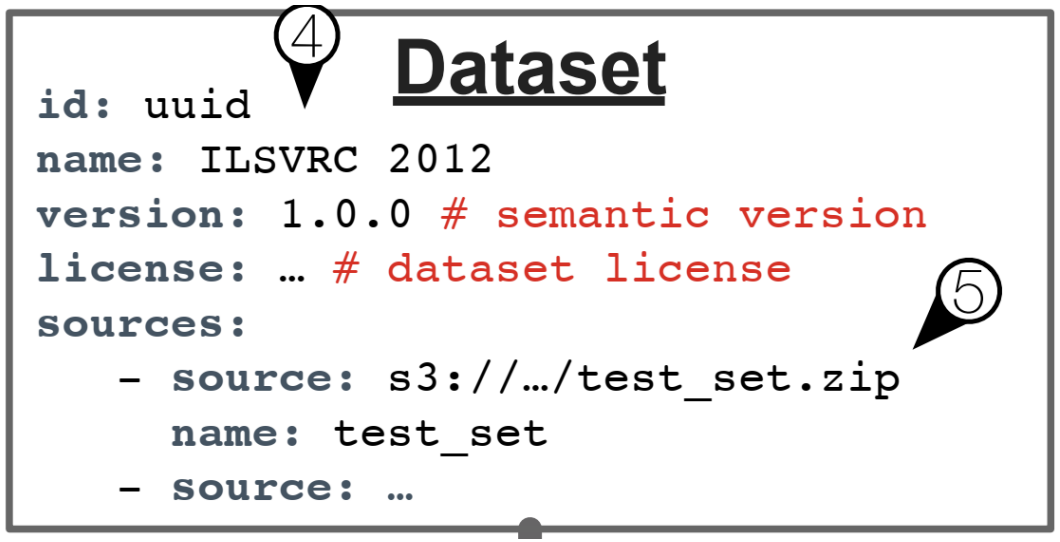
Software Manifest

- Defines the software environment for a DL task
- All executions occur within the specified container
- Specified environment variables are setup after running the container



Dataset Manifest

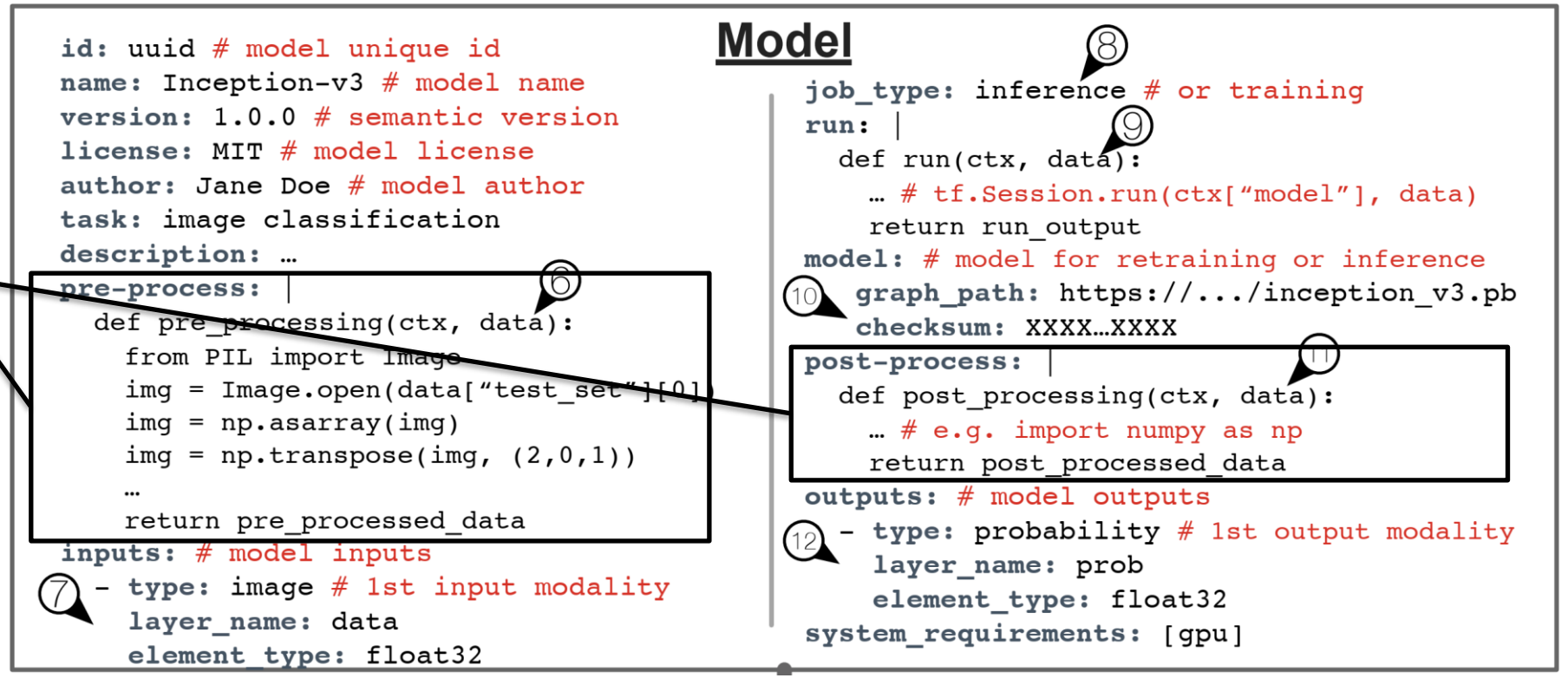
- Defines the training, validation, or test dataset
- The source location defines where to download the dataset from



Model Manifest

- Defines the logic to run a DL task and the required artifact sources

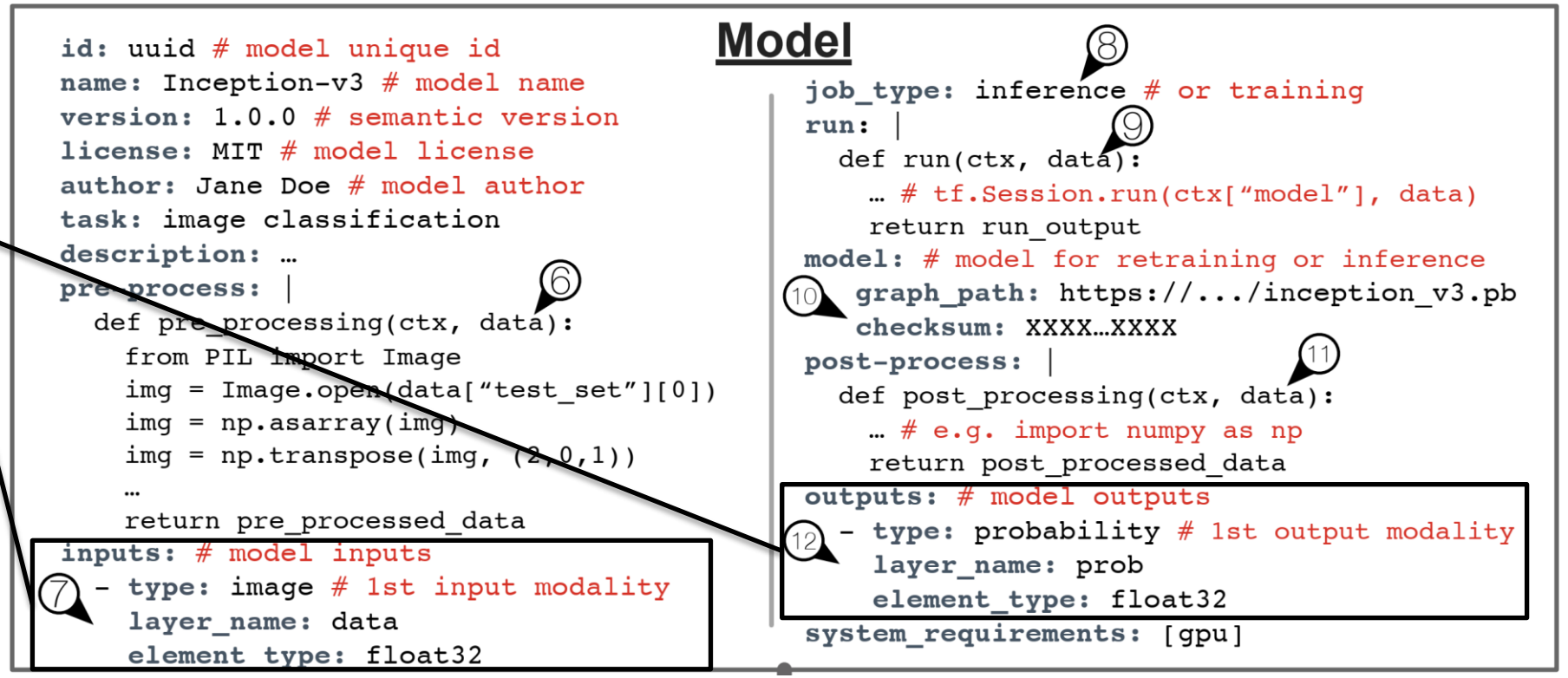
Python functions, executed by the runtime through the Python sub-interpreter



Model Manifest

- Defines the logic to run a DL task and the required artifact sources

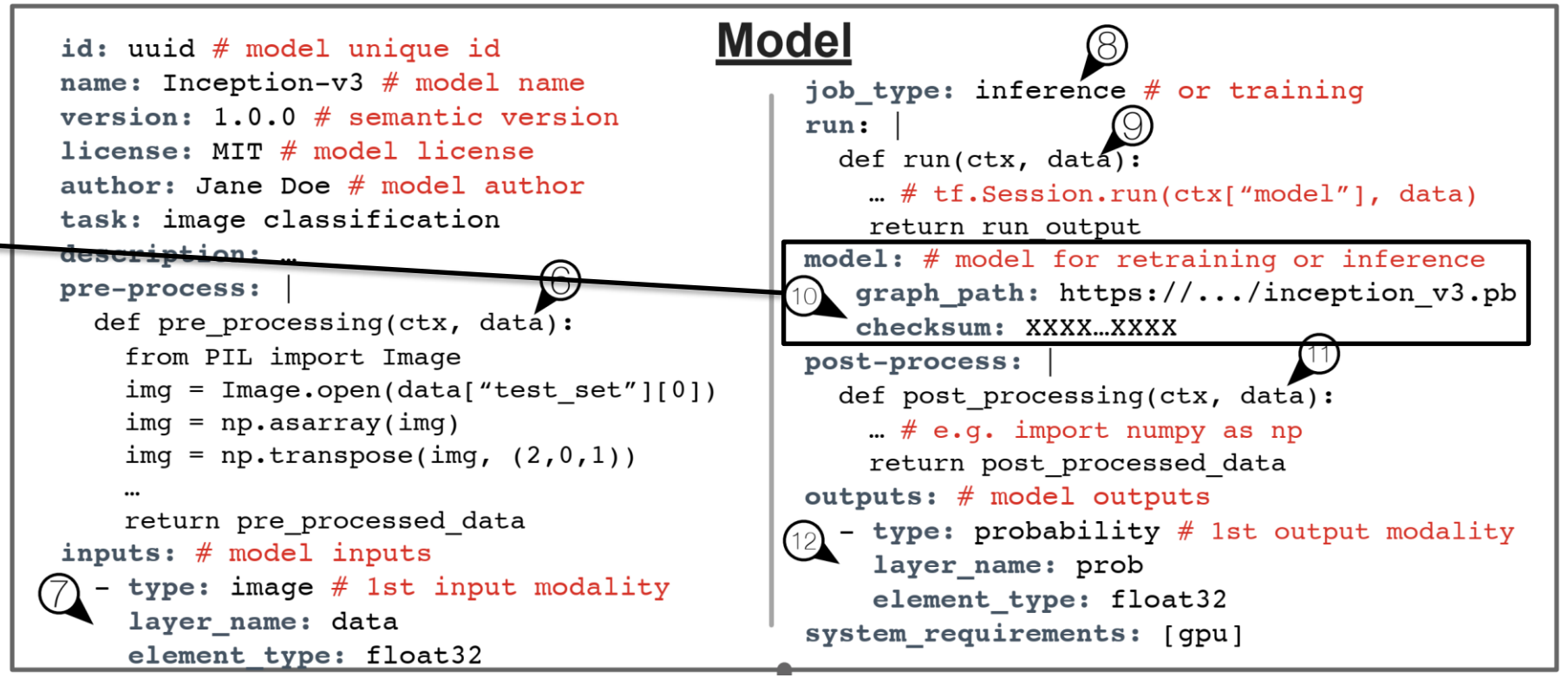
Input and output formats



Model Manifest

- Defines the logic to run a DL task and the required artifact sources

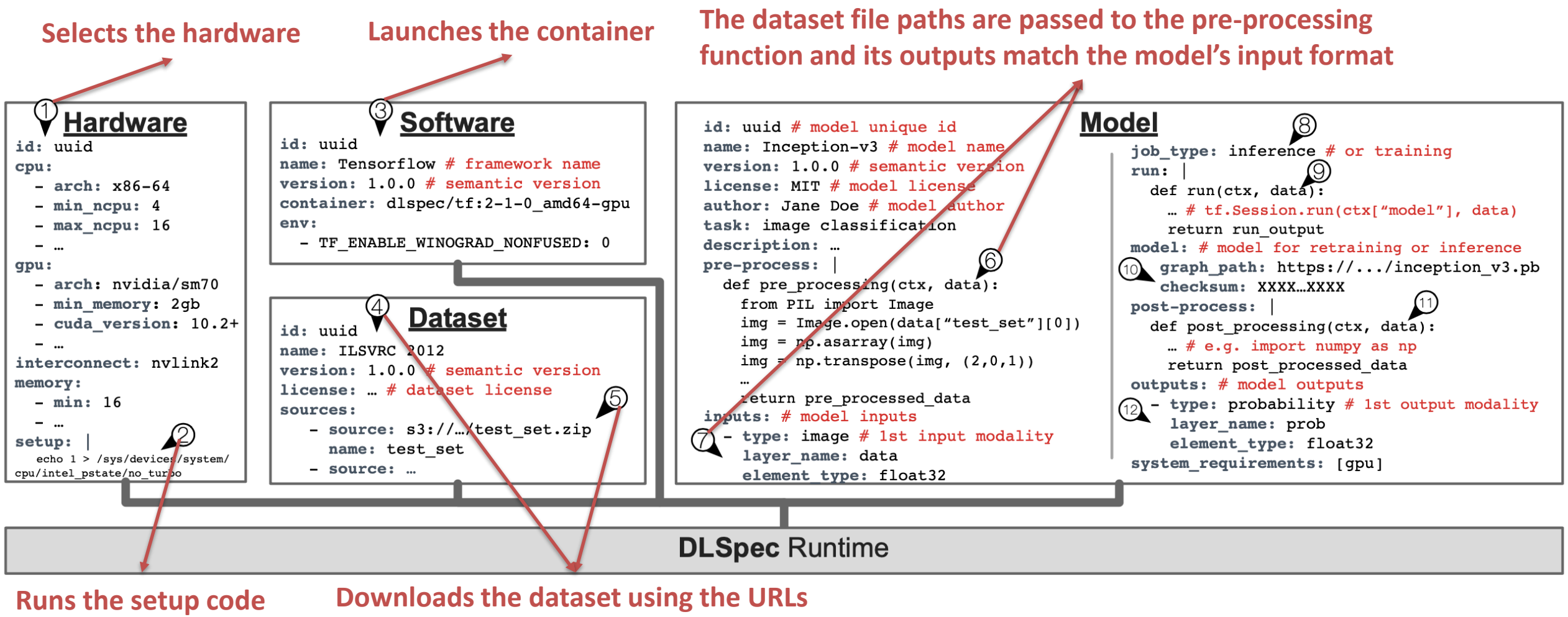
Remote resources hosted on FTP, HTTP, or file servers



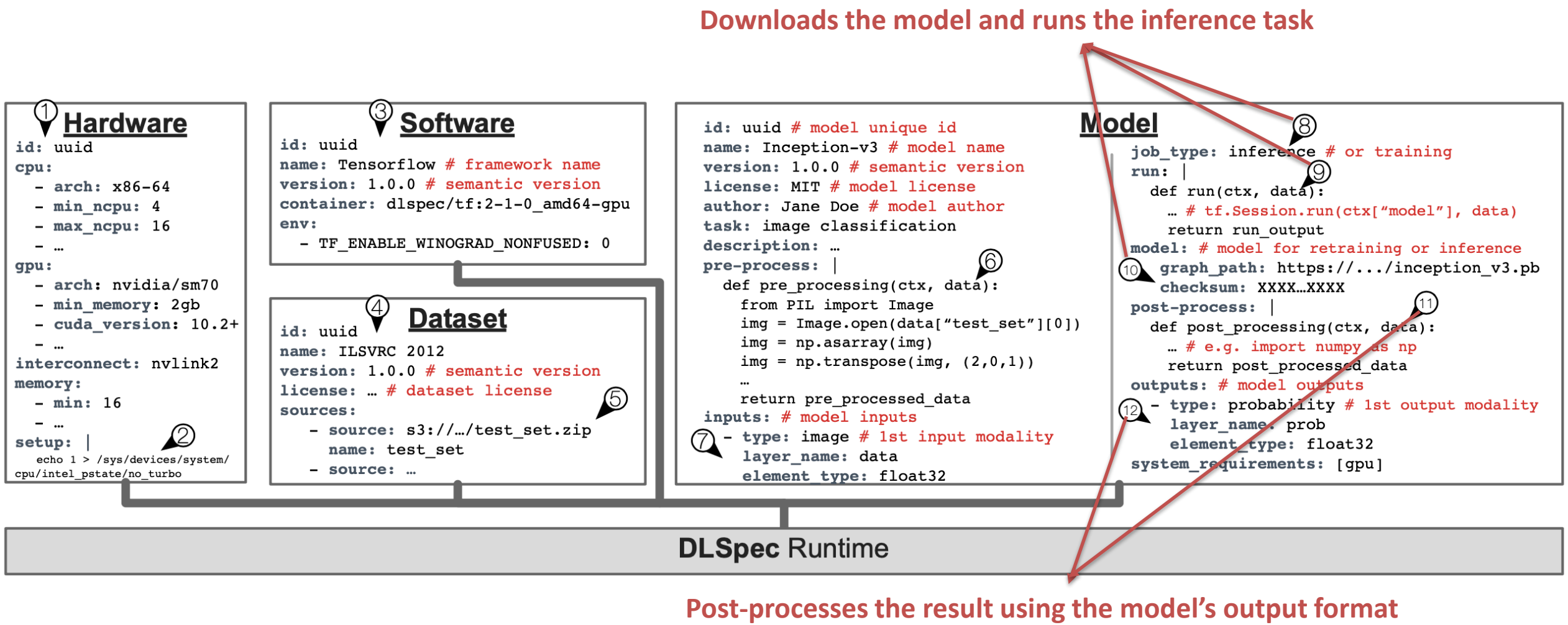
Reference Log

- A text file provided by the specification author for others to refer to. It contains:
  - IDs of the manifests used to create it
  - Achieved accuracy/performance on DL task
  - Expected outputs
  - Author-specified information (e.g. hyper-parameters used in training)

A DLSpec Runtime Consumes the Manifests



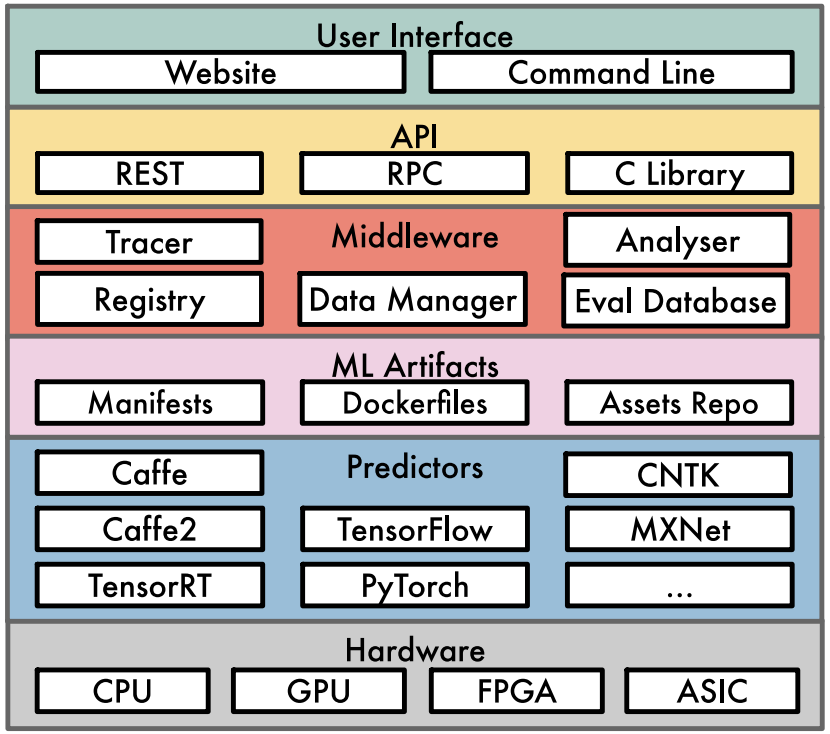
A DLSpec Runtime Consumes the Manifests



A Runtime for Benchmarking DL Inference - MLModelScope

The Design and Implementation of a Scalable DL Benchmarking Platform, IEEE CLOUD'20

- A distributed runtime that consumes the DLSpec for inference
  - Web and command line UI
  - Middleware, e.g. registry, database, tracer
  - Framework agents
  - Other modular components



Conclusion

- An exchange specification, such as DLSpec, enables a streamlined way to share, reproduce, and compare DL tasks
- DLSpec takes the first step in defining a DL task for both training and inference and captures the different aspects of DL model reproducibility
- We are actively working on refining the specifications as new DL tasks are introduced

Thank you

Abdul Dakkak[1*], Cheng Li[1*], Jinjun Xiong[2], Wen-mei Hwu[1]

University of Illinois Urbana-Champaign[1], IBM Research[2]