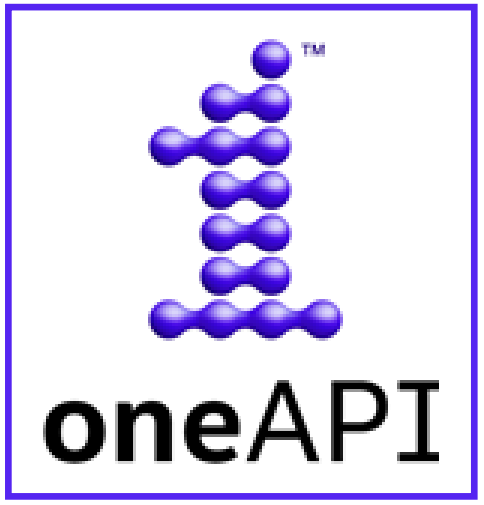


oneAPI Data Analytics Library

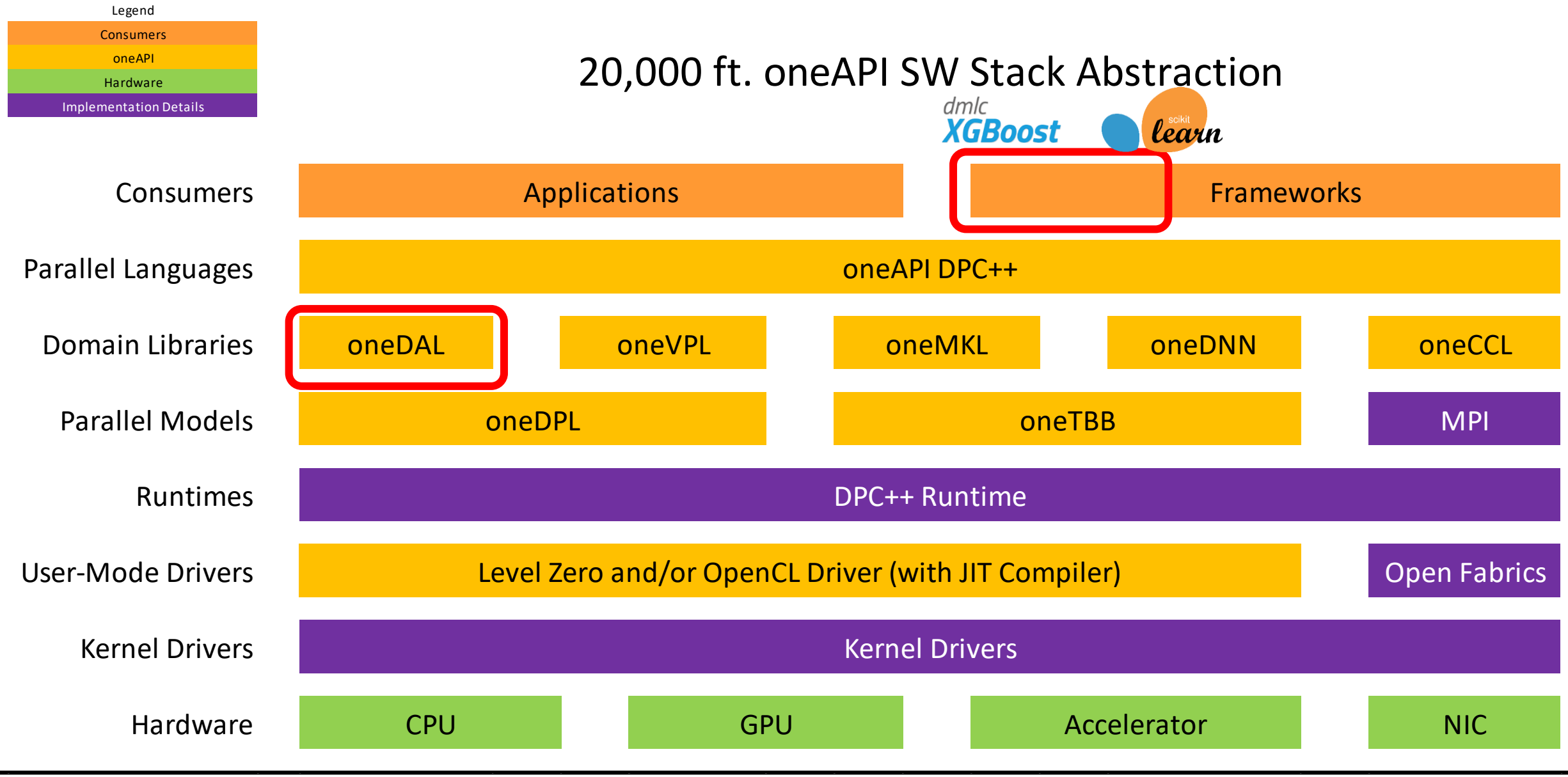
oneAPI industry specification

- Specifies interfaces
- Does not prescribe implementation choices, including underlying threading models or specific accelerator support

Intel’s oneAPI product

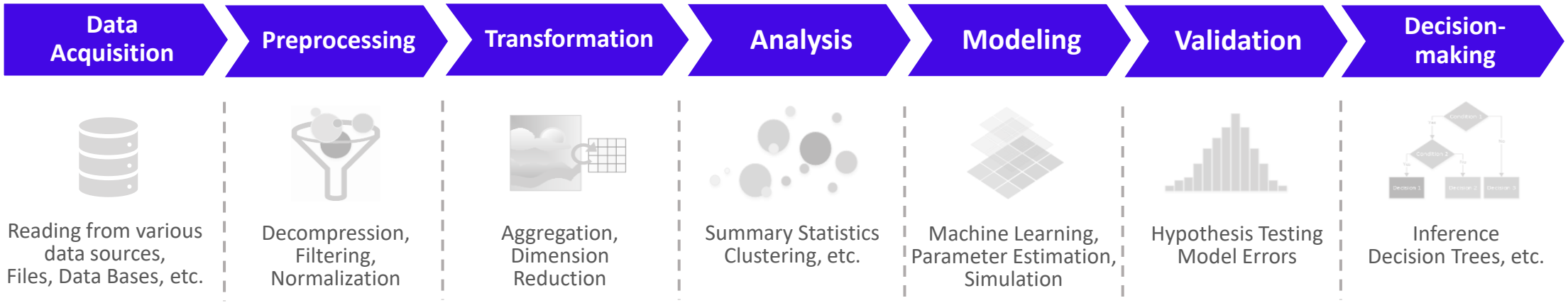
- Is an implementation of the oneAPI specification
- Makes choices about optimization techniques and implementation details
- Broader set of functionality for other interface languages such as Python/Java

oneAPI specification | Intel’s implementation of oneDAL


Legend
Consumers
oneAPI
Hardware
Implementation Details
20,000 ft. oneAPI SW Stack Abstraction
dmlc XGBoost
scikit learn
Consumers
Applications
Frameworks
Parallel Languages
oneAPI DPC++
Domain Libraries
oneDAL
oneVPL
oneMKL
oneDNN
oneCCL
Parallel Models
oneDPL
oneTBB
MPI
Runtimes
DPC++ Runtime
User-Mode Drivers
Level Zero and/or OpenCL Driver (with JIT Compiler)
Open Fabrics
Kernel Drivers
Kernel Drivers
Hardware
CPU
GPU
Accelerator
NIC

Overview

The oneAPI Data Analytics Library (oneDAL) is a collection of highly optimized algorithmic building blocks for all stages of data analytics.



oneDAL features:

- API designed to be hardware and vendor independent
- Relies on C++17 and DPC++
- Support for batch, distributed and online modes
- Support for heterogenous and sparse data

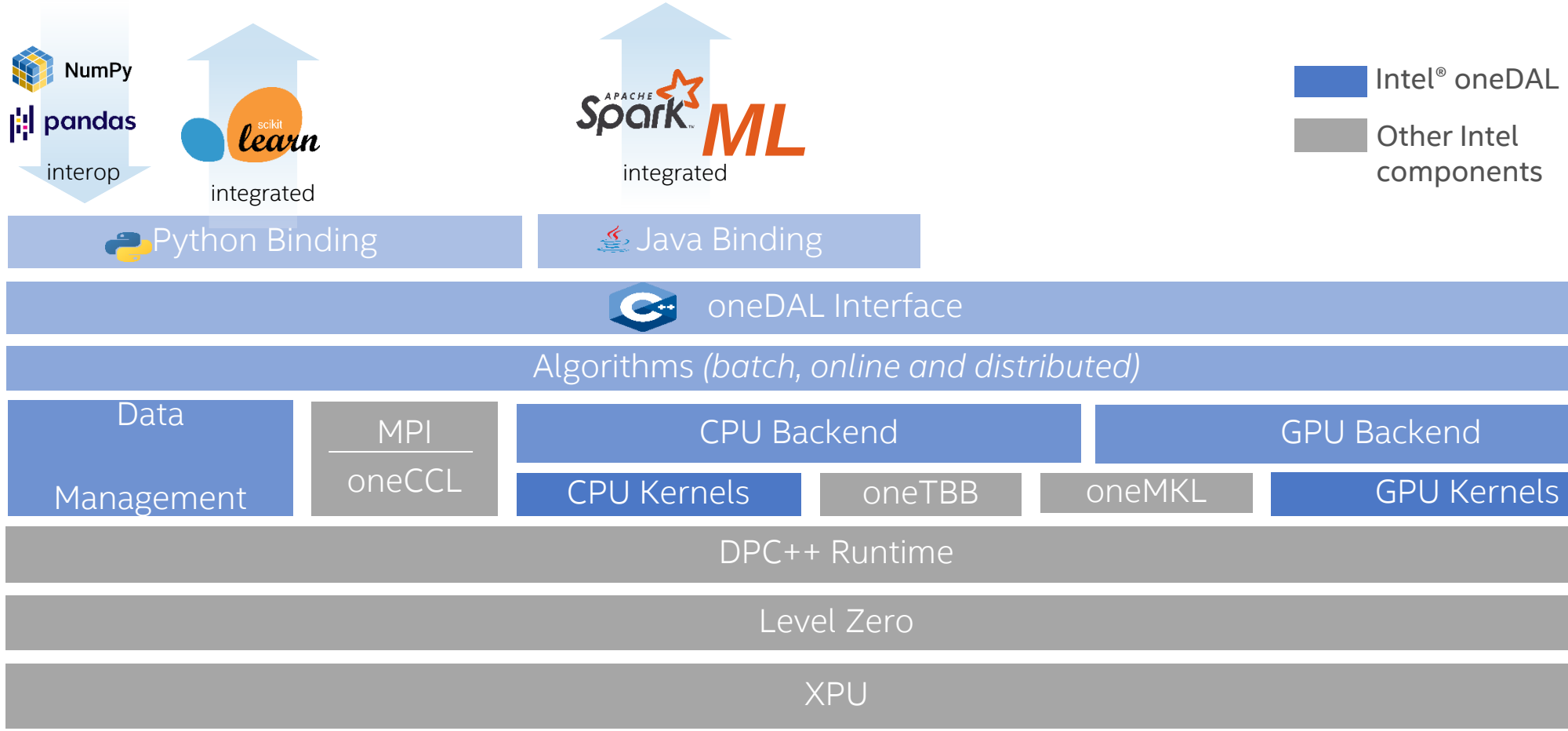
Examples of oneDAL algorithms:

K-Means	kNN	PCA	…
Data sources	Table	Accessors	….

oneAPI specification

Links: GitHub Specification Documentation

Intel® oneDAL

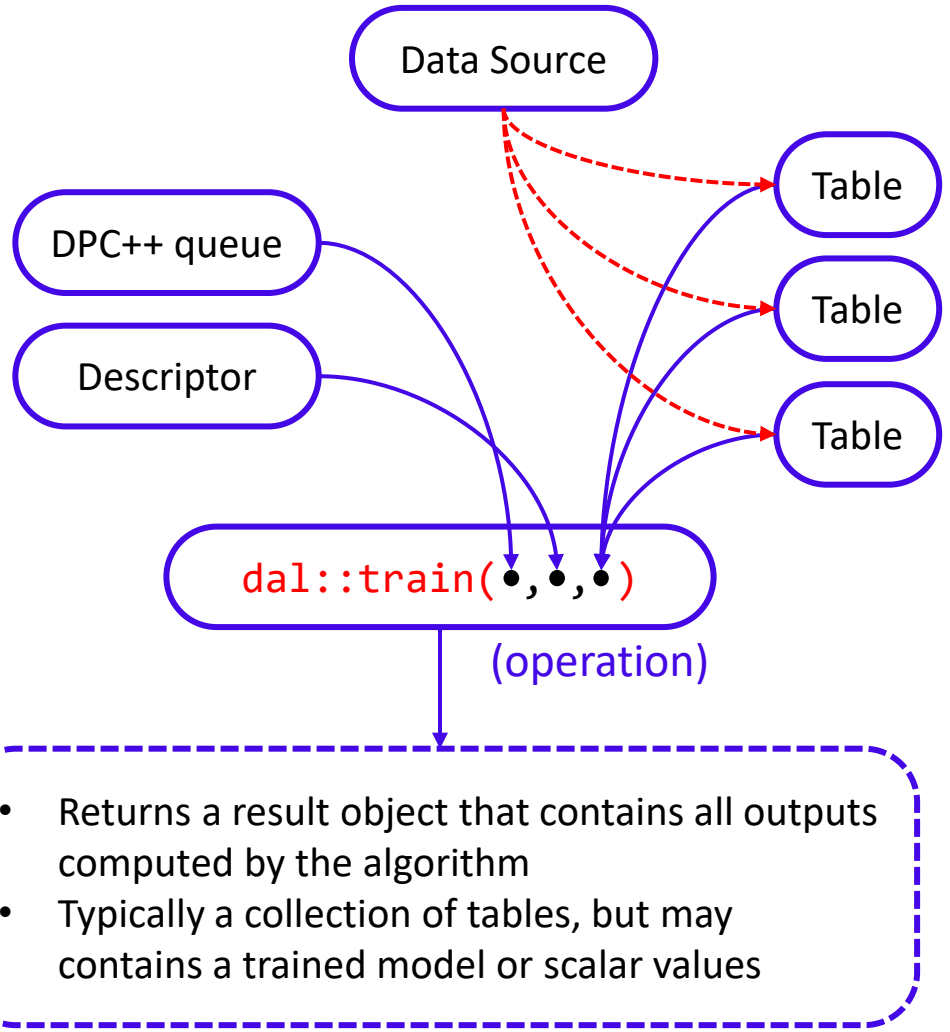


Intel's implementation of oneDAL

oneDAL Programming Model

Key abstractions

Data Source	Out-of-memory storage, extract datasets from external sources such as databases, files, remote storage
Table	In-memory numerical data organized in a tabular view with several rows and columns
Accessor	Provides unified access to the data from Table, acquire data in a desired fomat
Descriptor	Represents an algorithm including all its meta-parameters
Operation	Free function that executes all algorithm's computational routines



oneDAL Example

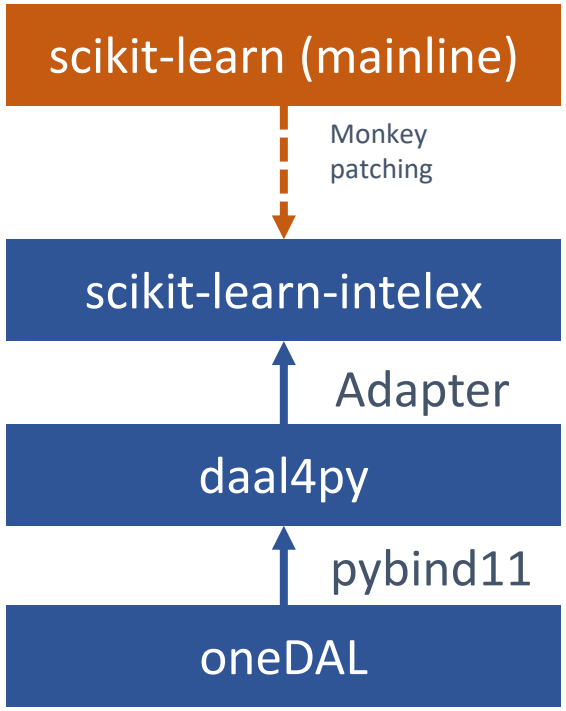
```
/* Include the following header that makes all oneDAL declarations available */
#include "oneapi/dal.hpp"

using namespace oneapi;

/* Create a DPC++ queue with the desired device selector. */
/* In this case, GPU selector is used */
const auto queue = sycl::queue{ sycl::gpu_selector{} };

/* Use CSV data source to read the data from the CSV file into a table */
const auto data = dal::read<dal::table>(queue, dal::csv::data_source{"data.csv"});

/* Create a PCA descriptor, configure its parameters, /*
/* and run the training algorithm on the data loaded from CSV. */
const auto pca_desc = dal::pca::descriptor<float>{}
   .set_component_count(3)
   .set_deterministic(true);

const dal::pca::train_result train_res = dal::train(queue, pca_desc, data);

/* Use the trained model for inference to reduce dimensionality of the data */
const dal::pca::model model = train_res.get_model();

const dal::table data_transformed =
   dal::infer(queue, pca_desc, model, data).get_transformed_data();
```

Example shows a typical workflow of using oneDAL algorithm on GPU. The example is provided for Principal Components Analysis (PCA) algorithm.

The following steps depict how to:

- Read the data from CSV file
- Run the training and inference operations for PCA

Intel(R) Extension for Scikit-learn



Same Code, Same Behavior

PASSED

- Scikit-learn, not scikit-learn-*like*
- Scikit-learn conformance (mathematical equivalence) defined by Scikit-learn
- Ongoing work with maintainers (INRIA) on perf. hotspots identification
- Auto-fallback in stock Scikit-learn for non-optimized functions

Anaconda

PyPI

oneAPI

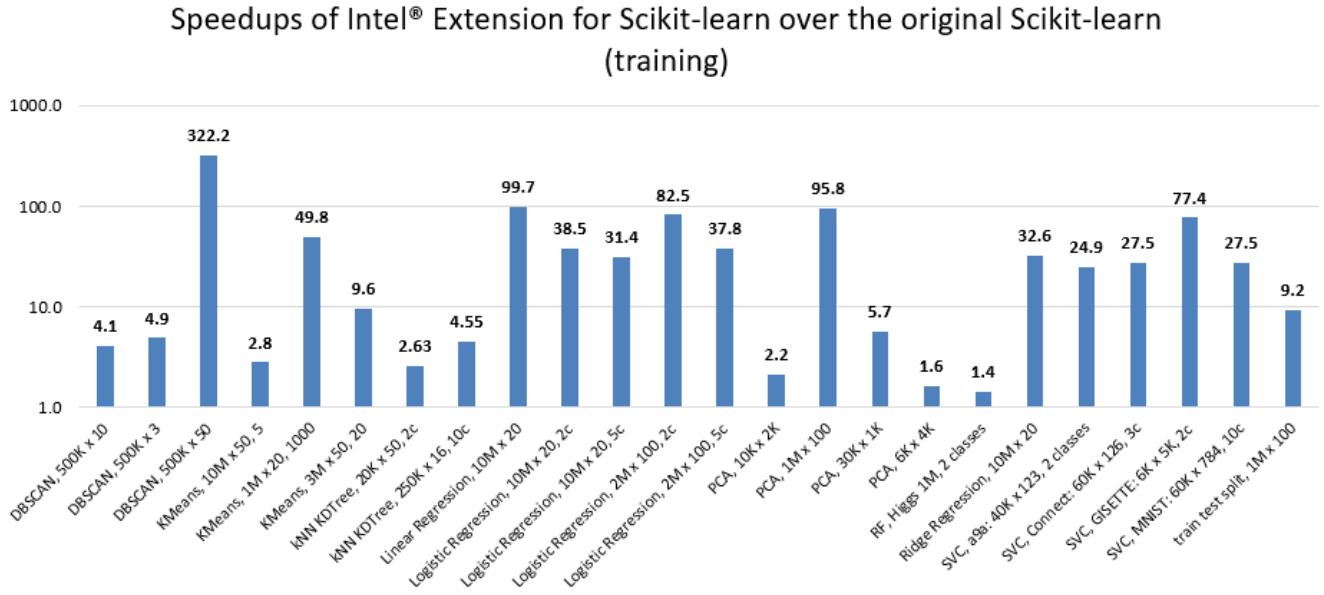
Intel GPU

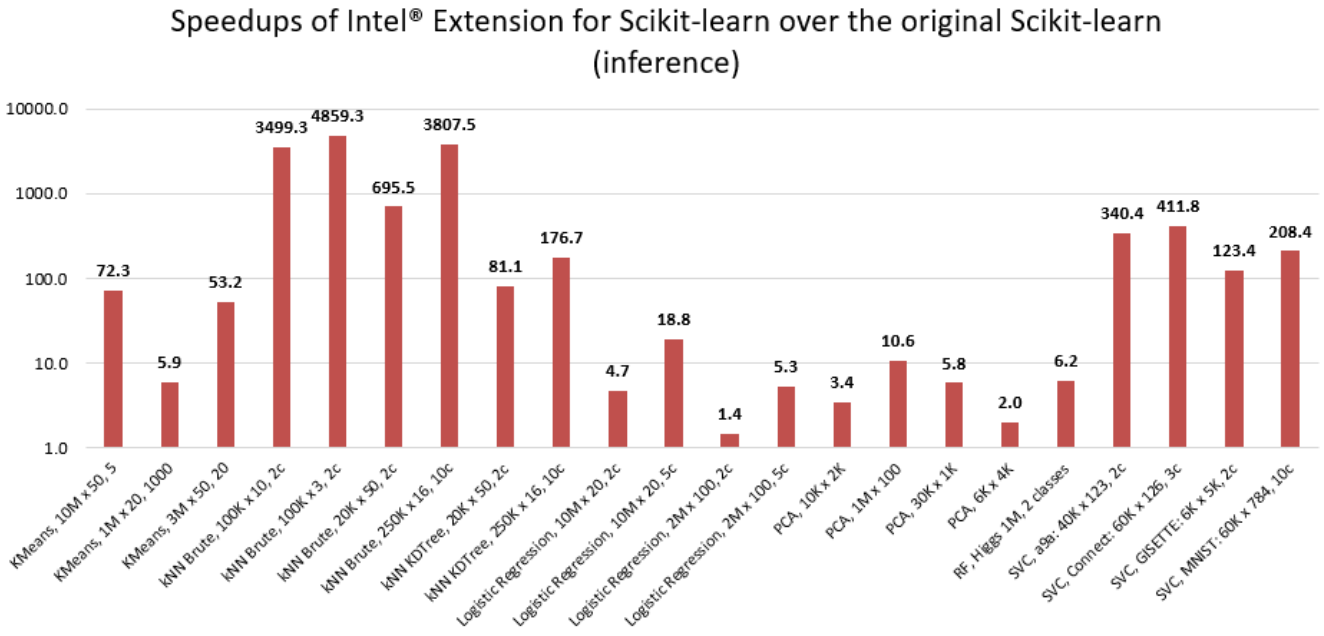
```
From sklearnex import patch_sklearn
patch_sklearn()
import dpctl
from sklearn.svm import SVC

X, Y = get_dataset()

with dpctl.device_context("gpu"):
    clf = SVC().fit(X, y)
    res = clf.predict(X)
```

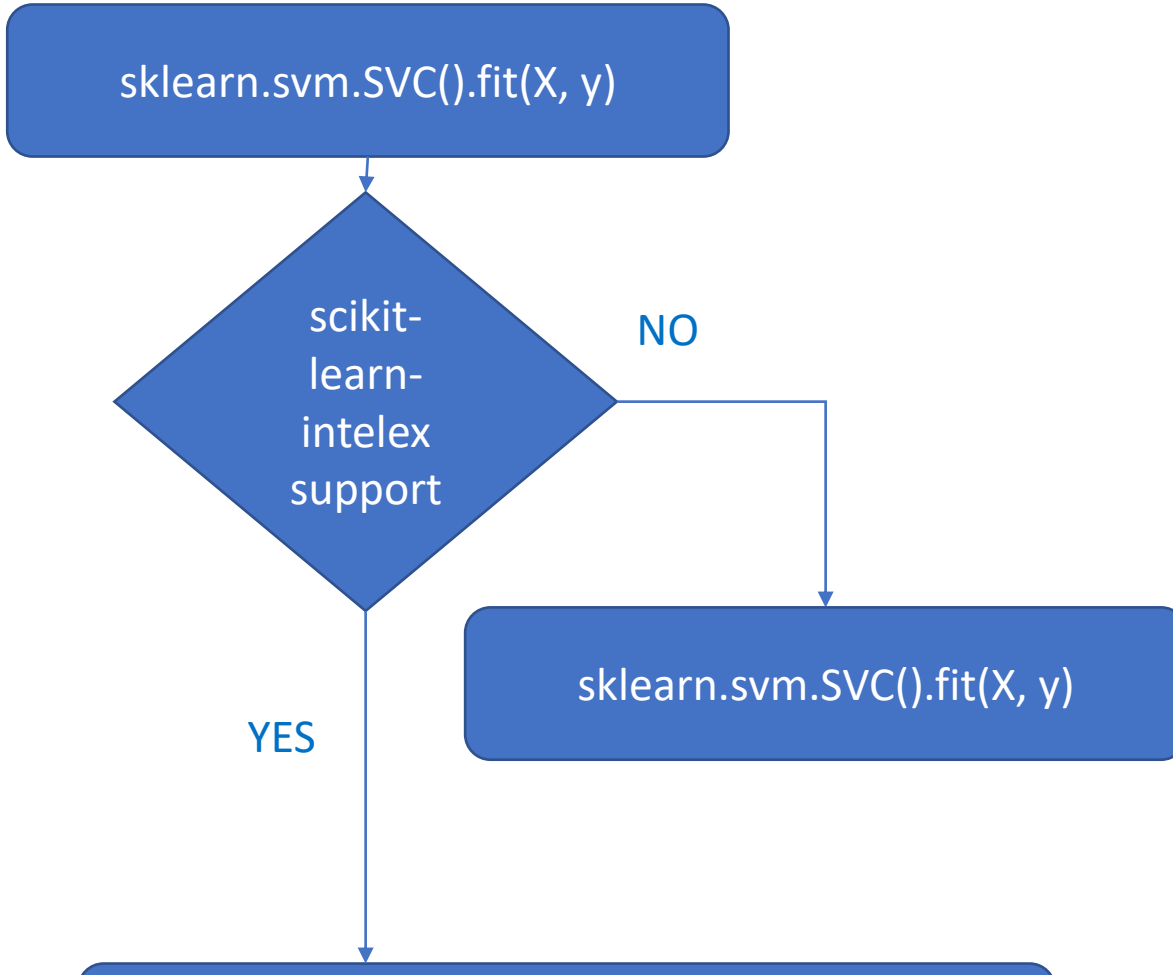
https://github.com/intel/scikit-learn-intelex

Intel's implementation

Intel(R) Extension for Scikit-learn

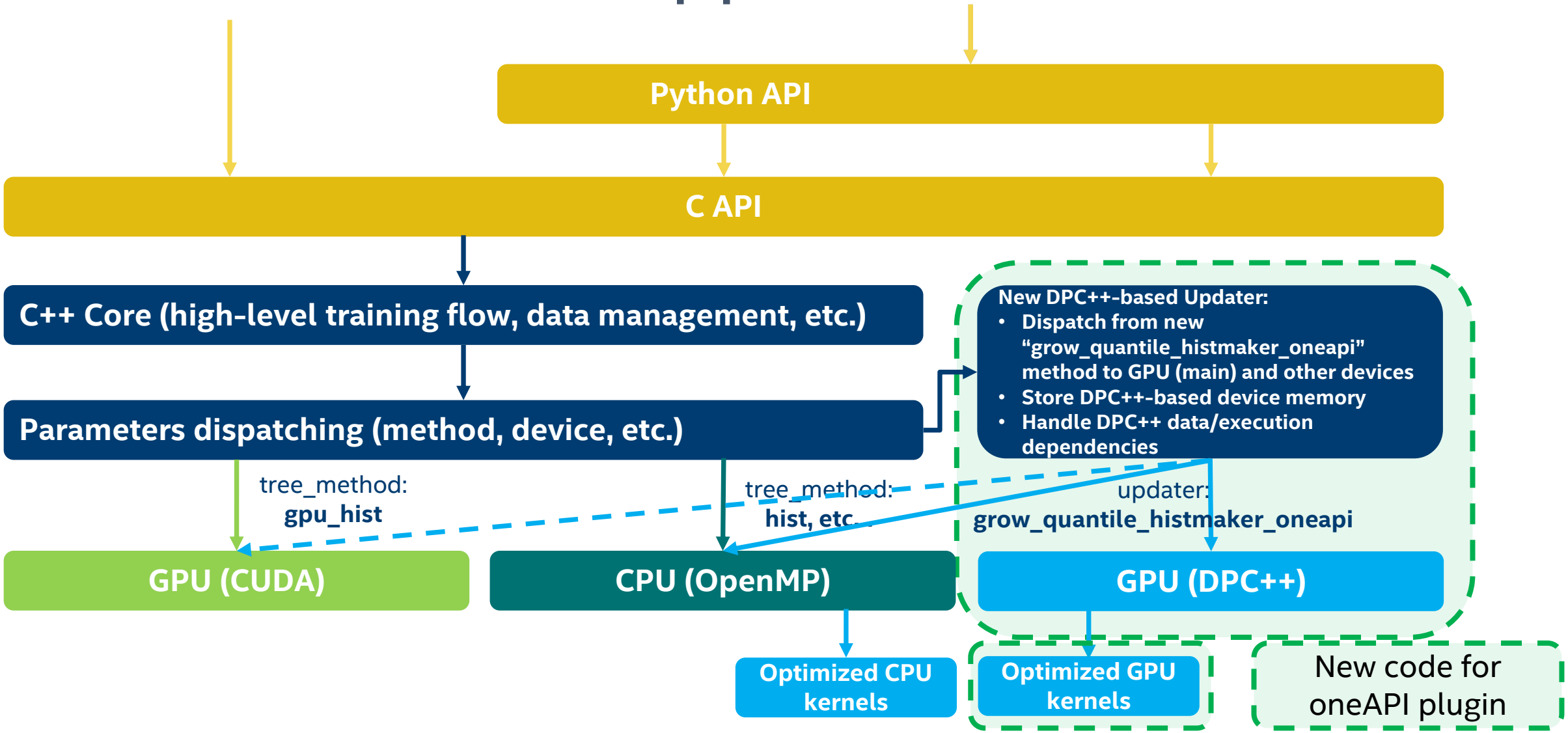






Intel's implementation

XGBoost oneAPI support

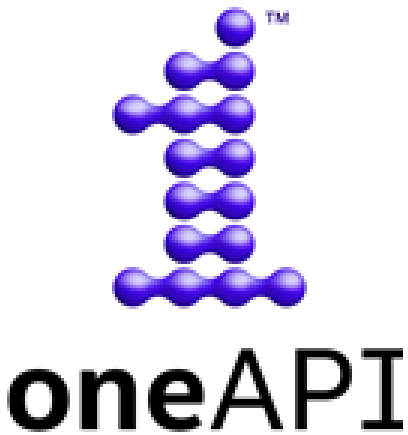


https://github.com/dmlc/xgboost/tree/master/plugin/updater_oneapi

Intel's implementation

Contacts

- Submit issues/PRs directly for oneDAL
  - https://github.com/oneapi-src/oneDAL
- Submit issues/PRs directly for Intel Extension for Scikit-learn
  - https://github.com/intel/scikit-learn-intelex
- Contribute to oneDAL specification
  - https://spec.oneapi.com/versions/latest/elements/oneDAL/source/index.html
- Contact us directly
  - nikolay.a.petrov@intel.com
  - onedal.maintainers@intel.com



Thank You!

http://oneapi.com