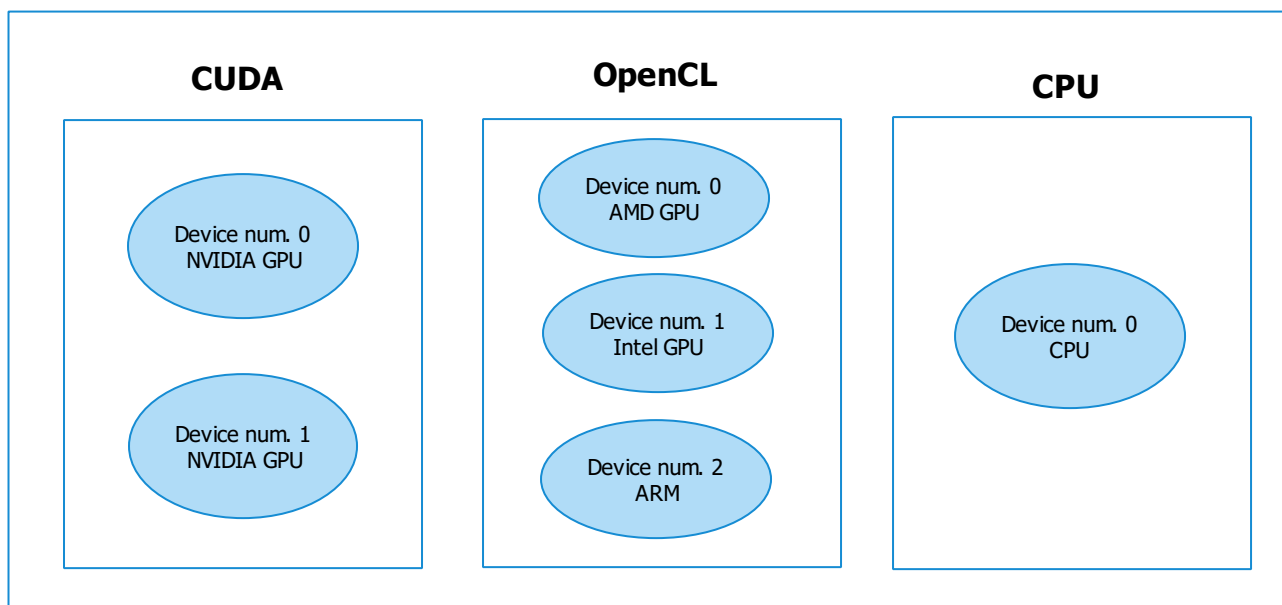


Hardware acceleration - environment setup

In this document, we illustrate how to setup an environment for using different types of hardware in your calculations. The CPU is the default calculation environment, beside this, MatDeck supports CUDA-capable NVIDIA GPUs, OpenCL devices including AMD GPUs/APUs, Intel Xeon Phi co-processors, mobile OpenCL devices like ARM and Qualcomm.

Control of the active environment on which the calculations will be made is achieved using a hierarchy of backends and devices that are supported in each backend (as shown on picture bellow).

Backends



Using the function `afp_supported_backends`, we get the list of all supported backends that can be used for calculations. In our case, calculations can be made on the CPU, using OpenCL or CUDA framework.

```
afp_supported_backends() = [ "cpu"  
                            "opengl"  
                            "cuda" ]
```

The default environment for calculations is the CPU, we can change the current environment with the function `afp_set_backend` and check which environment is currently in use with the `afp_backend` function. When we setup an active environment, we change the global properties and the selected environment will stay active even if we close MatDeck. Keep this in mind when you restart your MatDeck application, it won't change the active backend in default state.

```
afp_set_backend("opengl") = true  
afp_backend() = "opengl"
```

In each environment, there can be several devices which support calculations within it. To check the number of devices which support calculations in the current environment use the function `afp_get_device_count`, and use the functions `afp_get_device` and `afp_set_device` to check/change the currently active device.

```
afp_get_device_count( ) = 3
```

```
afp_get_device( ) = 1
```

```
afp_set_device(1) = true
```

To display the information about currently selected device, use the function `afp_device_info`.

```
afp_device_info( ) = [ "Intel(R)_HD_Graphics_620"  
                      "OpenCL"  
                      "Intel(R) OpenCL"  
                      "2.1" ]
```

Each time you change the active device, use the `afp_device_info()` function afterward to be certain that you activated the device that you planned to.

This was a brief overview of the architecture that stands behind the usage of CUDA or OpenCL on the GPU in your calculations.