# Used special libraries and types

**Boost**

This library is an union of libraries giving lot of settings like a toolbox. In our code, we use this library to provide us special behaviour and types such as locker, mutex, pointers.

**Mutex**

A Mutex is an object that can be locked because the behaviour of this function of variable is sensitive and can cause security issues, that is why we use locker of mutex.

**Locker**

```
boost::unique_lock<boost::mutex> lock( mutex );
```

We can see that the lock is a template function coming from boost library.
In fact we specify a mutex to lock.
A mutex can be locked only by one locker that's also why we use unique_locker.
The unique locker is a thread that throw exception is something happen.

**Thread**

The threads are a process that is happening next to the others.
When we talk about multi_thread, it means that multiple independent processes are running in the same time.

**Pointers**

Here we will talk about shared pointer coming from boost library.
Shared pointer is a template class that is used to store a pointer to a dynamically allocated object.
Like this the pointed object AND the pointer are destroyed after the process, not only the object or the pointer.

**PCL**

Our library to manage almost everything. It is optimised and open source but not only. It provides lot of functionalities but we can wonder that every point cloud libraries on internet provide such functionalities.

In fact this PCL library is one of the most easy to use and there is a reason, the types of data.
In PCL there are a lot of variable types that are available to makes the code easy and readable.
And this is the same process for functions.
With this library lot of things are convertible but we can recall some notions about ptr, clouds.

**pcl::PointCloud<pcl::PointXYZ>::Ptr**

These pointer are pointer to a point cloud template class, and in our case it use point in 3D as type.
This pointer follow a logic, it allow to reducer the size, the overload of the system and the copy.
In fact this is a pointer to a point cloud object defined by the type into the <>.

**Pointcloud**

This can be represented as a 3D array, each columns representing a dimension X Y or Z and each values the position. Each lines represent then a point in 3D.

We can have point clouds of type XYZRGBA , this is coloured point clouds with R G B and transparency component, so the number off letter indicate the dimension of this point cloud, here we have 7 dimensions.

When we use only point of type XYZ these are white points, so all the values settled in RGBA are 255.

**Special Variable Type**

HRESULT

This return type is always here in c++ as soon as we don't use another one. This is a typical behaviour of c++ that is some kind of Boolean indicating the good behaviour of a function.
HRESULT is always or almost always a return type.
When the process have no errors to throw then it return S_OK.

We can see this type of variable as a boolean component.