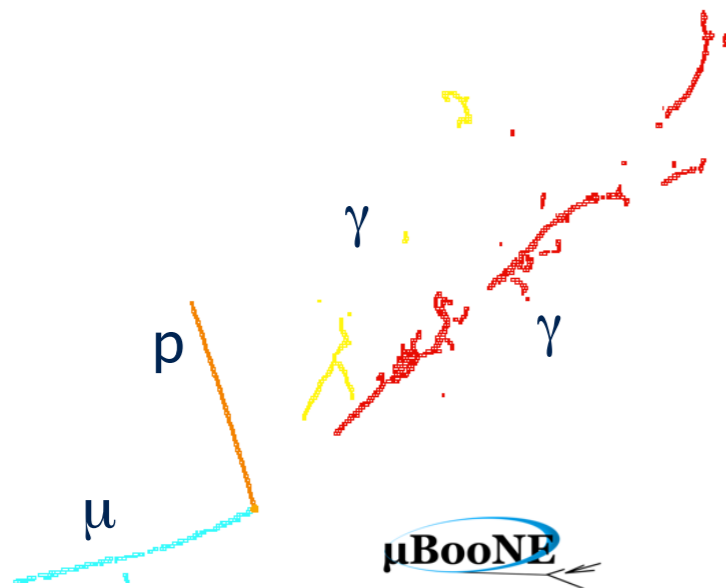




Pandora Exercise 2: A New Algorithm



J. S. Marshall for the Pandora Team
MicroBooNE Pandora Workshop
July 11-14th 2016, Cambridge





Getting Started

First step is to setup the Pandora standalone development environment.

Input events must be in Pandora persistency formats, so you can either use:

1. Events written out in LArSoft during exercise 1, or
2. The library of example events provided for workshop [here](#) (docdb username/password)

Pre-requisites: C++11 compiler, cmake, ROOT installation including TEVE libraries.

Can you run the following? `root $ROOTSYS/tutorials/eve/calorimeters.C`

 or equivalent location on your system

These instructions have been tested with:

-Scientific Linux CERN SLC release 6.7, gcc 4.9.3, ROOT 5.34.32

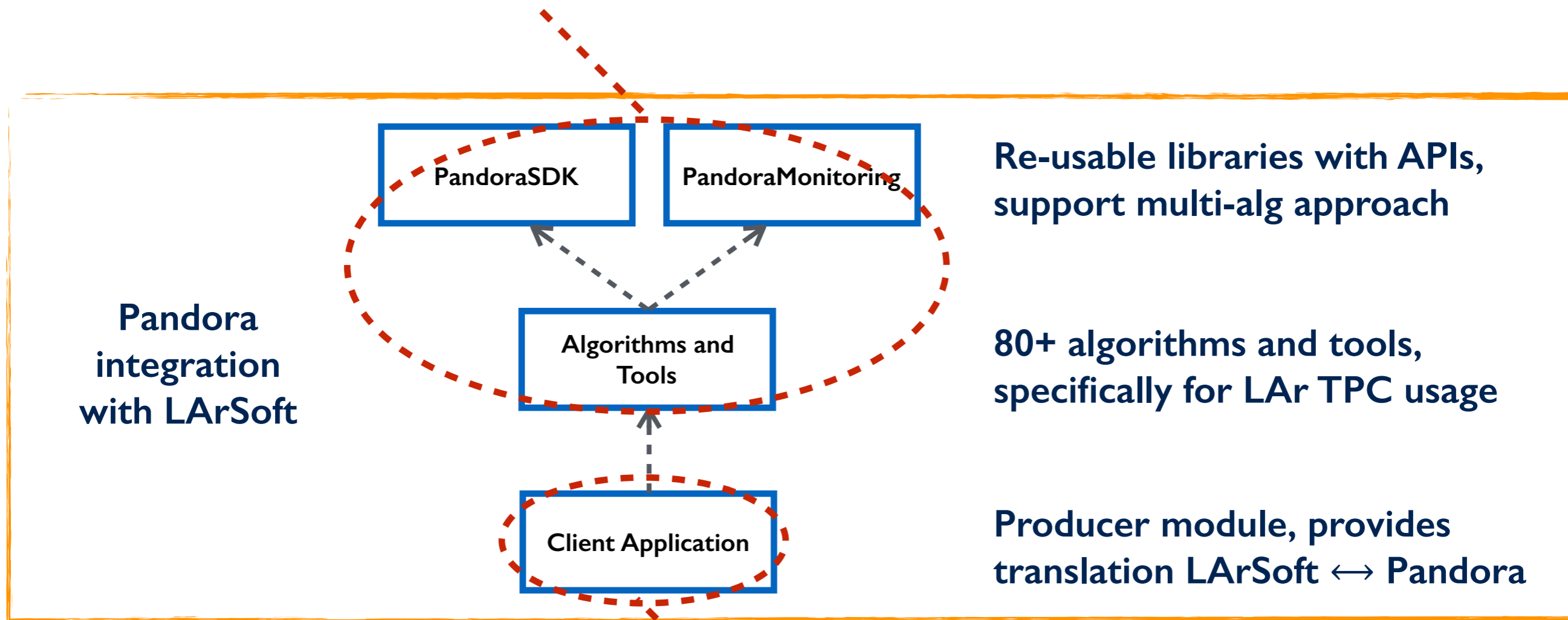
-Ubuntu 14.04.4 LTS, gcc 4.8.4, ROOT 5.34.32

-OSX El Capitan, 10.11.5, Apple LLVM 7.0.2 (clang-700.1.81), ROOT 5.34.32



Pandora Products

Step 1: Build common Pandora products and LAr TPC algorithms



Step 2: Build application and library tailored to this workshop



Step I: Common Pandora Products



Recommendation is to use Pandora CMake build system, although can use simple Makefiles for a more 'hands-on' approach.

First build core Pandora libraries we will need: PandoraSDK, PandoraMonitoring and all the LAr TPC reconstruction algorithms in the LArContent library.

```
export MY_TEST_AREA=/path/to/your/test/area
export ROOT_CMAKE_MODULE_PATH=/path/to/your/FindROOT.cmake/file
export PANDORA_PFA_VERSION=v02-08-03

cd $MY_TEST_AREA
git clone https://github.com/PandoraPFA/PandoraPFA

cd PandoraPFA
git checkout $PANDORA_PFA_VERSION

mkdir build
cd build

cmake -DCMAKE_MODULE_PATH=$ROOT_CMAKE_MODULE_PATH -DPANDORA_MONITORING=ON -DPANDORA_LAR_CONTENT=ON ..
make -j4 install
```

See screen grabs on next few slides...



Step I: Common Pandora Products



```
cd $MY_TEST_AREA
git clone https://github.com/PandoraPFA/PandoraPFA

cd PandoraPFA
git checkout $PANDORA_PFA_VERSION
```

```
bash-3.2$ export MY_TEST_AREA=`pwd`
bash-3.2$ export ROOT_CMAKE_MODULE_PATH=/opt/local/etc/root5/cmake/
bash-3.2$ export PANDORA_PFA_VERSION=v02-08-03
bash-3.2$ git clone https://github.com/PandoraPFA/PandoraPFA

Cloning into 'PandoraPFA'...
remote: Counting objects: 10121, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10121 (delta 3), reused 0 (delta 0), pack-reused 10113
Receiving objects: 100% (10121/10121), 1.94 MiB | 880.00 KiB/s, done.
Resolving deltas: 100% (7887/7887), done.
Checking connectivity... done.
bash-3.2$ cd PandoraPFA/
bash-3.2$ git checkout $PANDORA_PFA_VERSION
Note: checking out 'v02-08-03'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-b` with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

HEAD is now at 7f702a2... Set package versions.



Step I: Common Pandora Products



```
mkdir build
```

```
cd build
```

```
cmake -DCMAKE_MODULE_PATH=$ROOT_CMAKE_MODULE_PATH -DPANDORA_MONITORING=ON -DPANDORA_LAR_CONTENT=ON ..
```

```
bash-3.2$ mkdir build
bash-3.2$ cd build
bash-3.2$ cmake -DCMAKE_MODULE_PATH=$ROOT_CMAKE_MODULE_PATH -DPANDORA_MONITORING=ON -DPANDORA_LAR_CONTENT=ON ..
-- The C compiler identification is AppleClang 7.0.2.7000181
-- The CXX compiler identification is AppleClang 7.0.2.7000181
-- Check for working C compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/cc
-- Check for working C compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/c++
-- Check for working CXX compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- PANDORA_SDK: ON
-- PANDORA_MONITORING: ON
-- LAR_PANDORA_CONTENT: ON
-- LC_PANDORA_CONTENT: OFF
-- EXAMPLE_PANDORA_CONTENT: OFF
-- INSTALL_DOC: OFF
--
-- -----
-- Change values with: cmake -D<Variable>=<Value>
-- CMAKE_INSTALL_PREFIX = /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA
-- CMAKE_BUILD_TYPE = RelWithDebInfo
-- INSTALL_DOC = OFF
-- CMAKE_MODULE_PATH =
--   /opt/local/etc/root5/cmake/;
--   /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/cmakemodules;
-- -----
--
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/build
```



Step I: Common Pandora Products



```
make -j4 install
```

```
-----
-- Change values with: cmake -D<Variable>=<Value>
-- CMAKE_INSTALL_PREFIX = /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA
-- CMAKE_BUILD_TYPE = RelWithDebInfo
-- INSTALL_DOC = OFF
-- CMAKE_MODULE_PATH =
--   /opt/local/etc/root5/cmake/;
--   /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/cmakemodules;
-----
```

```
-----
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/build/PandoraSDK-v02-03-00/src/PandoraSDK-build
[ 25%] Performing build step for 'PandoraSDK'
Scanning dependencies of target PandoraSDK
[ 1%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Api/PandoraApi.cc.o
[ 3%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Api/PandoraApiImpl.cc.o
[ 5%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Api/PandoraContentApi.cc.o
[ 7%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Api/PandoraContentApiImpl.cc.o
[ 9%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Helpers/ClusterFitHelper.cc.o
[10%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Helpers/MCParticleHelper.cc.o
[12%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Helpers/XMLHelper.cc.o
[14%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Managers/AlgorithmManager.cc.o
[16%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Managers/AlgorithmObjectManager.cc.o
[18%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Managers/CaloHitManager.cc.o
[20%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Managers/ClusterManager.cc.o
[21%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Managers/GeometryManager.cc.o
[23%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Managers/InputObjectManager.cc.o
[25%] Building CXX object CMakeFiles/PandoraSDK.dir/src/Managers/Manager.cc.o
```

```
-----
-- Change values with: cmake -D<Variable>=<Value>
-- CMAKE_INSTALL_PREFIX = /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA
-- CMAKE_BUILD_TYPE = RelWithDebInfo
-- INSTALL_DOC = OFF
-- CMAKE_MODULE_PATH =
--   /opt/local/etc/root5/cmake/;
--   /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/cmakemodules;
-----
```

```
-----
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/build/LArContent-v02-07-03/src/LArContent-build
[ 91%] Performing build step for 'LArContent'
Scanning dependencies of target LArContent
[ 0%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCheating/CheatingClusterCharacterisationAlgorithm.cc.o
[ 1%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCheating/CheatingClusterCreationAlgorithm.cc.o
[ 1%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCheating/CheatingCosmicRayIdentificationAlg.cc.o
[ 2%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCheating/CheatingCosmicRayShowerMatchingAlg.cc.o
[ 3%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCheating/CheatingEventSlicingTool.cc.o
[ 3%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCheating/CheatingNeutrinoCreationAlgorithm.cc.o
[ 4%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCheating/CheatingNeutrinoDaughterVerticesAlgorithm.cc.o
[ 5%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCheating/CheatingPfoCreationAlgorithm.cc.o
[ 5%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCheating/CheatingVertexCreationAlgorithm.cc.o
[ 6%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCustomParticles/CustomParticleCreationAlgorithm.cc.o
[ 7%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCustomParticles/ShowerParticleBuildingAlgorithm.cc.o
[ 7%] Building CXX object CMakeFiles/LArContent.dir/larpandoracontent/LArCustomParticles/TrackParticleBuildingAlgorithm.cc.o
```

```
-----
-- Change values with: cmake -D<Variable>=<Value>
-- CMAKE_INSTALL_PREFIX = /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA
-- CMAKE_BUILD_TYPE = RelWithDebInfo
-- INSTALL_DOC = OFF
-- CMAKE_MODULE_PATH =
--   /opt/local/etc/root5/cmake/;
--   /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/cmakemodules;
-----
```

```
-----
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/build/
[ 58%] Performing build step for 'PandoraMonitoring'
Scanning dependencies of target PandoraMonitoring
[ 25%] Building CXX object CMakeFiles/PandoraMonitoring.dir/src/PandoraMonitoring.cc.o
[ 50%] Building CXX object CMakeFiles/PandoraMonitoring.dir/src/PandoraMonitoringApi.cc.o
[ 75%] Building CXX object CMakeFiles/PandoraMonitoring.dir/src/TTreewrapper.cc.o
[100%] Linking CXX shared library lib/libPandoraMonitoring.dylib
[100%] Built target PandoraMonitoring
[ 62%] Performing install step for 'PandoraMonitoring'
[100%] Built target PandoraMonitoring
```



Step 2: Workshop Content

Next setup a small library and application for this workshop, alongside a set of generic Pandora learning exercises.

```
cd $MY_TEST_AREA
git clone https://github.com/PandoraPFA/WorkshopContent

cd WorkshopContent
mkdir build
cd build

cmake -DCMAKE_MODULE_PATH="$ROOT_CMAKE_MODULE_PATH;$MY_TEST_AREA/PandoraPFA/cmakemodules" \
      -DPandoraSDK_DIR=$MY_TEST_AREA/PandoraPFA \
      -DPANDORA_MONITORING=ON \
      -DPandoraMonitoring_DIR=$MY_TEST_AREA/PandoraPFA \
      -DLArContent_DIR=$MY_TEST_AREA/PandoraPFA ..

make -j4 install
```




Step 2: Workshop Content



```

cmake -DCMAKE_MODULE_PATH="$ROOT_CMAKE_MODULE_PATH;$MY_TEST_AREA/PandoraPFA/CMakeModules" -DPandoraSDK_DIR=$MY_TEST_AREA/PandoraPFA -DPANDORA_MONITORING=ON
-DPandoraMonitoring_DIR=$MY_TEST_AREA/PandoraPFA -DLArContent_DIR=$MY_TEST_AREA/PandoraPFA ..

-- The C compiler identification is AppleClang 7.0.2.7000181
-- The CXX compiler identification is AppleClang 7.0.2.7000181
-- Check for working C compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/cc
-- Check for working C compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/c++
-- Check for working CXX compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Check for PandoraSDK (02.03.00)
-- Check for PandoraSDK_LIBRARIES: PandoraSDK
-- Check for PandoraSDK_PANDORASDK_LIBRARY: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/lib/libPandoraSDK.dylib -- ok
-- Found PandoraSDK: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA (Required is at least version "02.03.00")
-- Check for LArContent (02.07.03)
-- Check for LArContent_LIBRARIES: LArContent
-- Check for LArContent_LARCONTENT_LIBRARY: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/lib/libLArContent.dylib -- ok
-- Found LArContent: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA (Required is at least version "02.07.03")
-- Check for PandoraMonitoring (02.03.00)
-- Check for PandoraMonitoring_LIBRARIES: PandoraMonitoring
-- Check for PandoraMonitoring_PANDORAMONITORING_LIBRARY: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/lib/libPandoraMonitoring.dylib -- ok
-- Found PandoraMonitoring: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA (Required is at least version "02.03.00")
-- Found ROOT: /opt/local/bin/root-config (Required is at least version "5.26.00")
-- Performing Test COMPILER_SUPPORTS_CXX_FLAGS
-- Performing Test COMPILER_SUPPORTS_CXX_FLAGS - Success
--
-----
-- Change values with: cmake -D<Variable>=<Value>
-- CMAKE_INSTALL_PREFIX = /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/workshopcontent
-- CMAKE_BUILD_TYPE = RelWithDebInfo
-- CMAKE_MODULE_PATH =
--   /opt/local/etc/root5/cmake/;
--   /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/CMakeModules;
--   /etc/cmake/;
--
-----
-- Check for PandoraSDK (02.03.00)
-- Check for PandoraSDK_LIBRARIES: PandoraSDK
-- Check for PandoraSDK_PANDORASDK_LIBRARY: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/lib/libPandoraSDK.dylib -- ok
-- Check for PandoraMonitoring (02.03.00)
-- Check for PandoraMonitoring_LIBRARIES: PandoraMonitoring
-- Check for PandoraMonitoring_PANDORAMONITORING_LIBRARY: /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/lib/libPandoraMonitoring.dylib -- ok
-- Performing Test COMPILER_SUPPORTS_CXX_FLAGS
-- Performing Test COMPILER_SUPPORTS_CXX_FLAGS - Success
--
-----
-- Change values with: cmake -D<Variable>=<Value>-- CMAKE_INSTALL_PREFIX = /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/workshopcontent
-- CMAKE_BUILD_TYPE = RelWithDebInfo
-- CMAKE_MODULE_PATH =
--   /opt/local/etc/root5/cmake/;
--   /Users/johnmarshall/Documents/WorkshopTest/PandoraPFA/CMakeModules;
--   /etc/cmake/;
--   /etc/cmake/;
--
-----
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/build

```



Step 2: Workshop Content



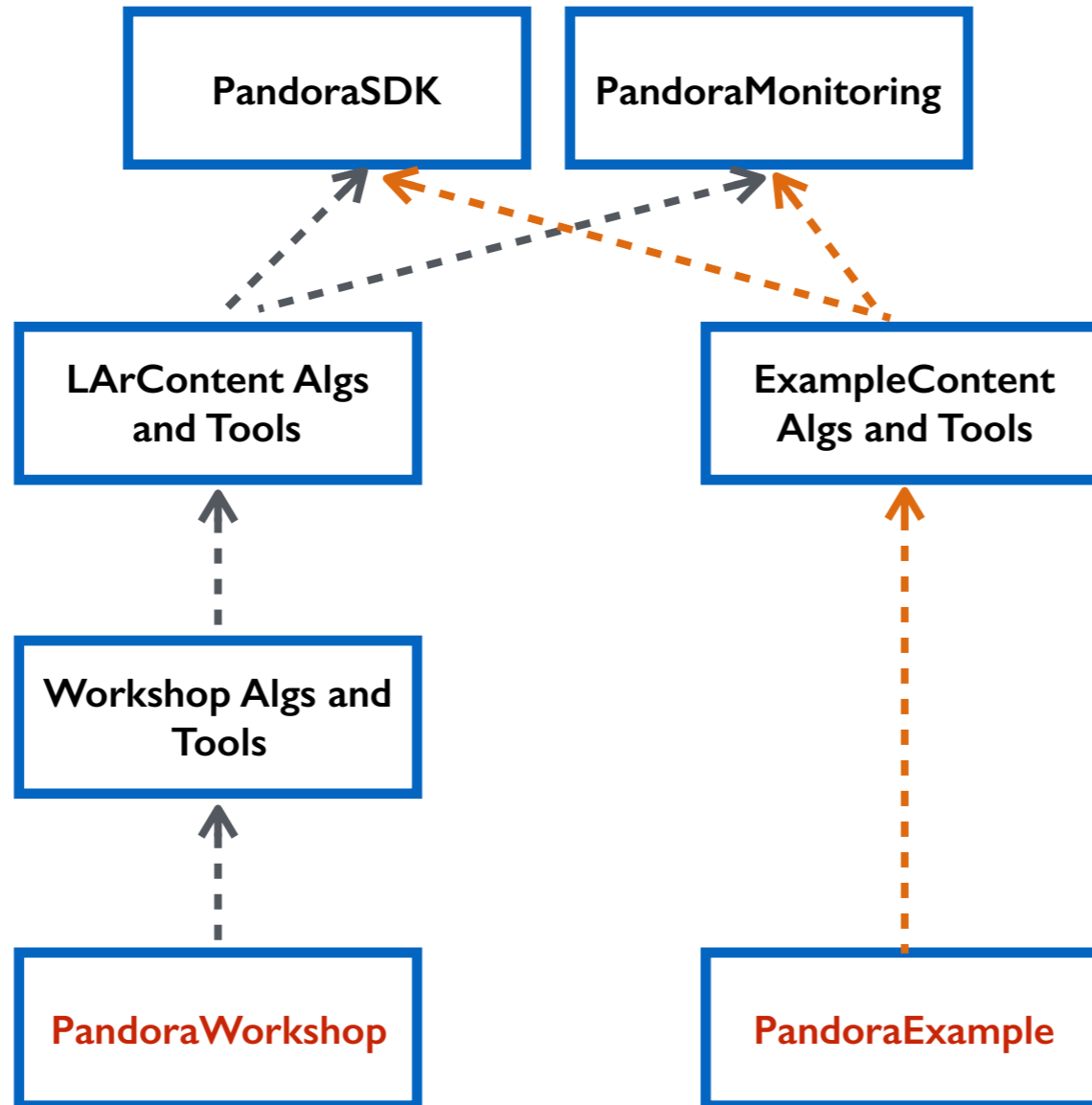
```

make install
Scanning dependencies of target WorkshopContent
[ 3%] Building CXX object workshopcontent/CMakeFiles/WorkshopContent.dir/Algorithms/TemplateAlgorithm.cc.o
[ 6%] Building CXX object workshopcontent/CMakeFiles/WorkshopContent.dir/Plugins/MicroBooNEPseudoLayerPlugin.cc.o
[ 10%] Building CXX object workshopcontent/CMakeFiles/WorkshopContent.dir/Plugins/MicroBooNETransformationPlugin.cc.o
[ 13%] Linking CXX shared library lib/libWorkshopContent.dylib
[ 13%] Built target WorkshopContent
Scanning dependencies of target PandoraWorkshop
[ 17%] Building CXX object workshopcontent/CMakeFiles/PandoraWorkshop.dir/Test/PandoraWorkshop.cc.o
[ 20%] Linking CXX executable bin/PandoraWorkshop
[ 20%] Built target PandoraWorkshop
Scanning dependencies of target ExampleContent
[ 24%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/AccessListsAlgorithm.cc.o
[ 27%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/CreateAdditionalCaloHitsAlgorithm.cc.o
[ 31%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/CreateClustersAlgorithm.cc.o
[ 34%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/CreateClustersDaughterAlgorithm.cc.o
[ 37%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/CreatePfosAlgorithm.cc.o
[ 41%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/CreateVerticesAlgorithm.cc.o
[ 44%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/DeleteClustersAlgorithm.cc.o
[ 48%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/DisplayListsAlgorithm.cc.o
[ 51%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/FragmentClustersAlgorithm.cc.o
[ 55%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/MergeClustersAlgorithm.cc.o
[ 58%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/ReconfigureClustersAlgorithm.cc.o
[ 62%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/SelectHitSubsetAlgorithm.cc.o
[ 65%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/UseAlgorithmToolAlgorithm.cc.o
[ 68%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/UsePluginsAlgorithm.cc.o
[ 72%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithms/WriteTreeAlgorithm.cc.o
[ 75%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleAlgorithmTools/ExampleAlgorithmTool.cc.o
[ 79%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExampleHelpers/ExampleHelper.cc.o
[ 82%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExamplePlugins/ExampleEnergyCorrectionPlugin.cc.o
[ 86%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExamplePlugins/ExampleParticleIdPlugin.cc.o
[ 89%] Building CXX object examplecontent/CMakeFiles/ExampleContent.dir/ExamplePlugins/ExamplePseudoLayerPlugin.cc.o
[ 93%] Linking CXX shared library lib/libExampleContent.dylib
[ 93%] Built target ExampleContent
Scanning dependencies of target PandoraExample
[ 96%] Building CXX object examplecontent/CMakeFiles/PandoraExample.dir/Test/PandoraExample.cc.o
[100%] Linking CXX executable bin/PandoraExample
[100%] Built target PandoraExample
Install the project...
-- Install configuration: "RelWithDebInfo"
-- Installing: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent
-- Installing: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Algorithms
-- Installing: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Algorithms/TemplateAlgorithm.h
-- Installing: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Plugins
-- Installing: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Plugins/MicroBooNEPseudoLayerPlugin.h
-- Installing: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Plugins/MicroBooNETransformationPlugin.h
-- Installing: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Test
-- Installing: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/examplecontent

```

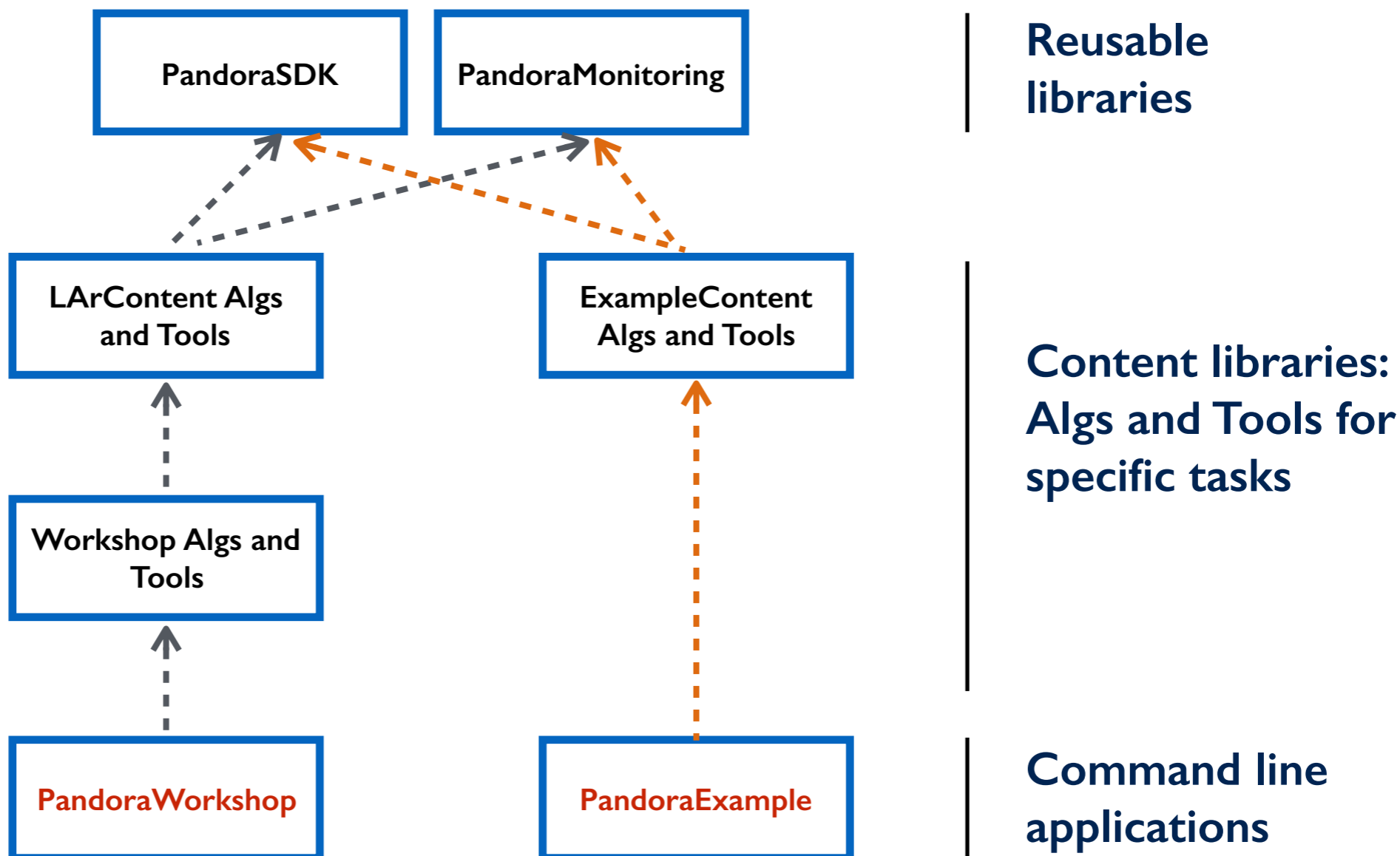


Workshop Test Area



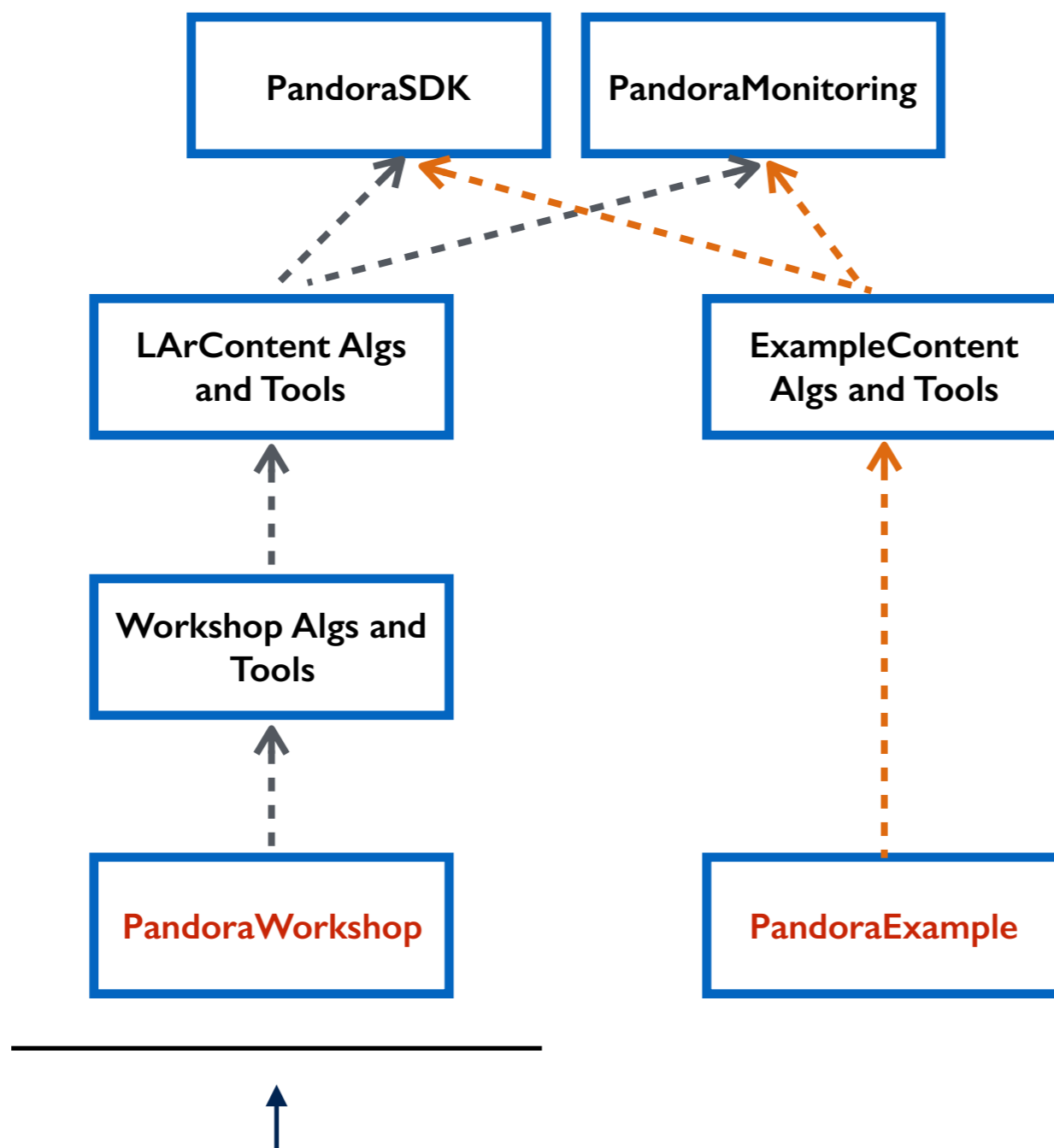


Workshop Test Area





Workshop Test Area



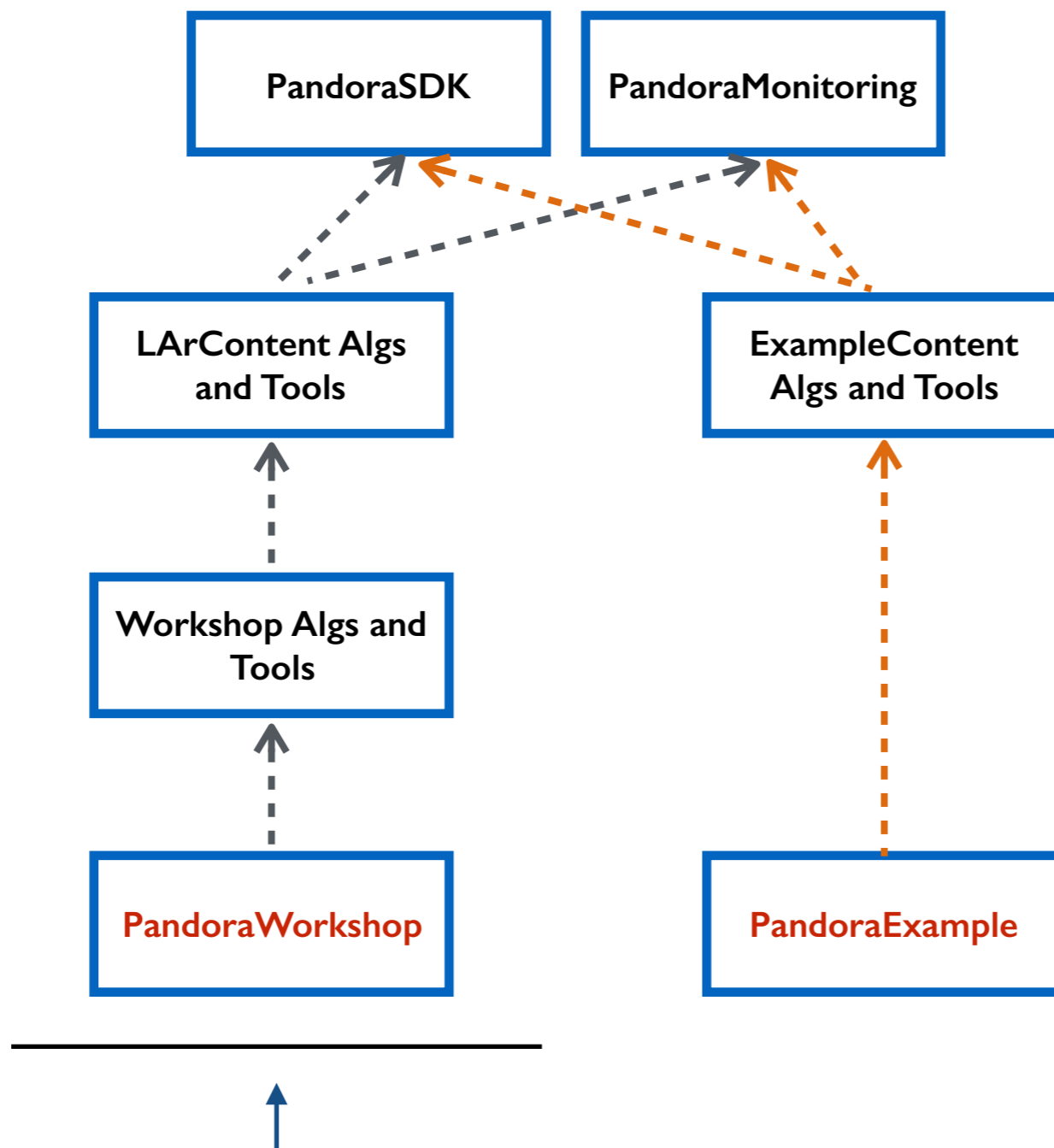
Main workshop development area, to which we will add new LAr-specific Algorithms and Tools

All LAr TPC Algorithms, Tools and Helper functions are accessible/reusable



Workshop Test Area

The LAr TPC Reconstruction

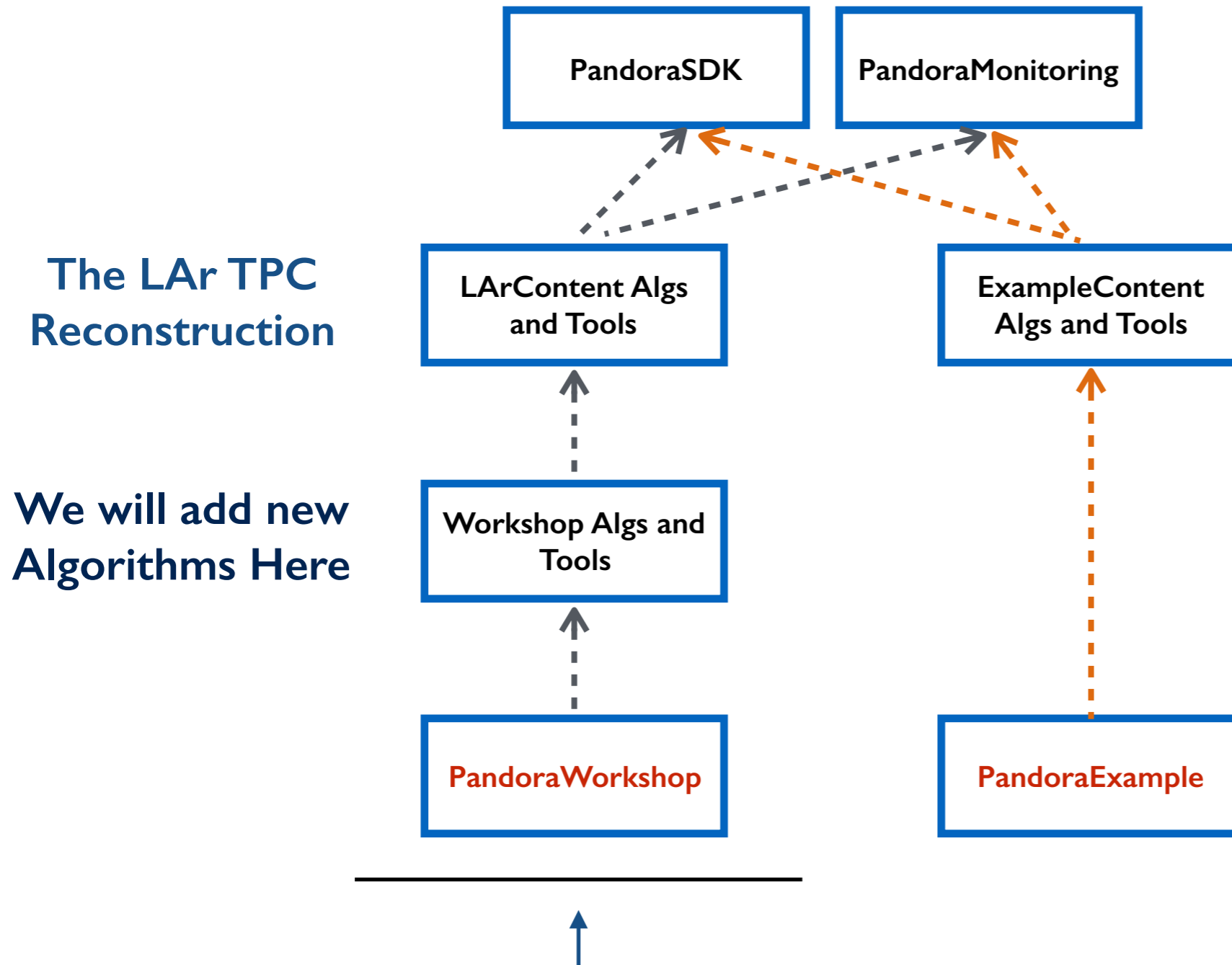


Main workshop development area, to which we will add new LAr-specific Algorithms and Tools

All LAr TPC Algorithms, Tools and Helper functions are accessible/reusable



Workshop Test Area



The LAr TPC Reconstruction

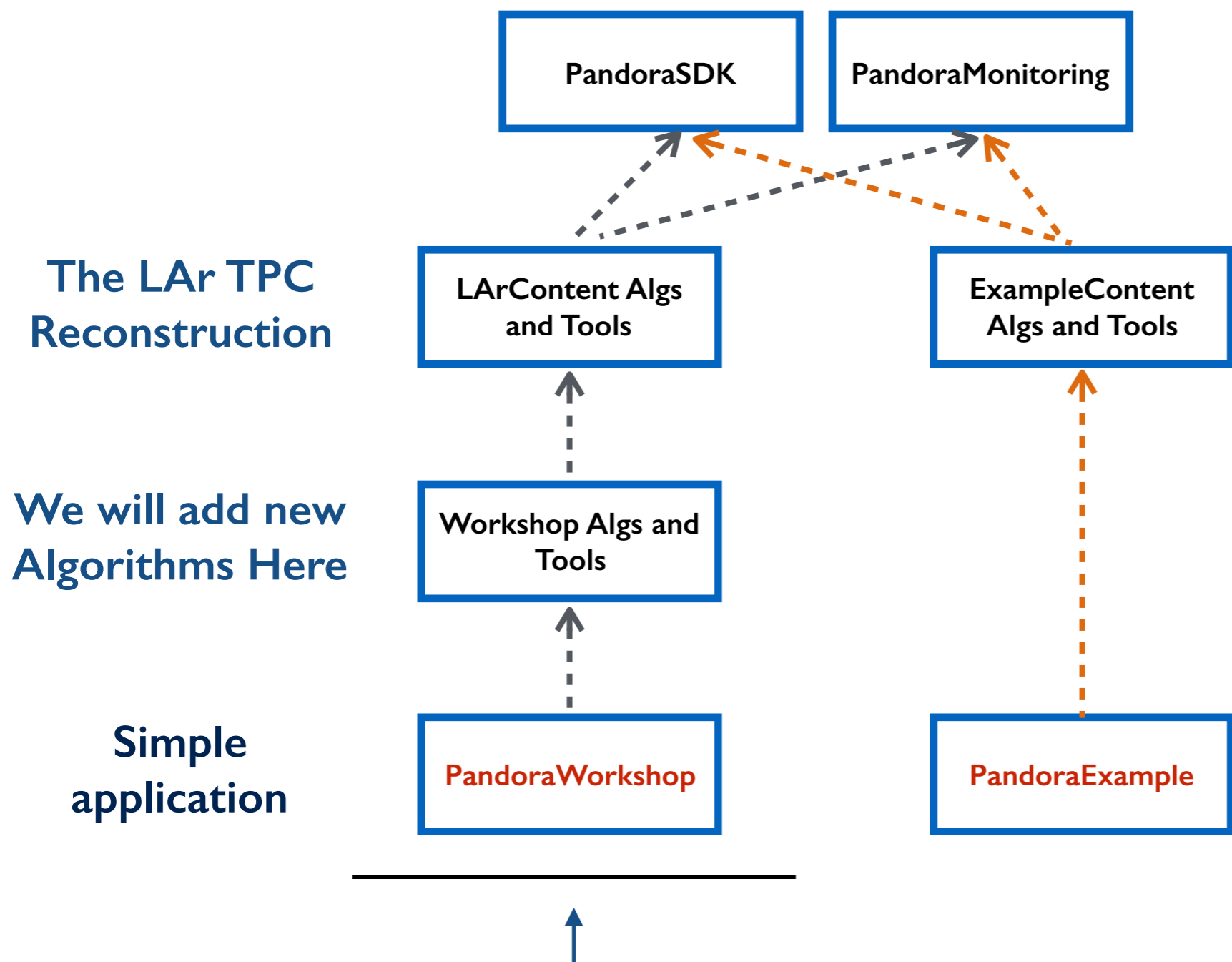
We will add new Algorithms Here

Main workshop development area, to which we will add new LAr-specific Algorithms and Tools

All LAr TPC Algorithms, Tools and Helper functions are accessible/reusable



Workshop Test Area

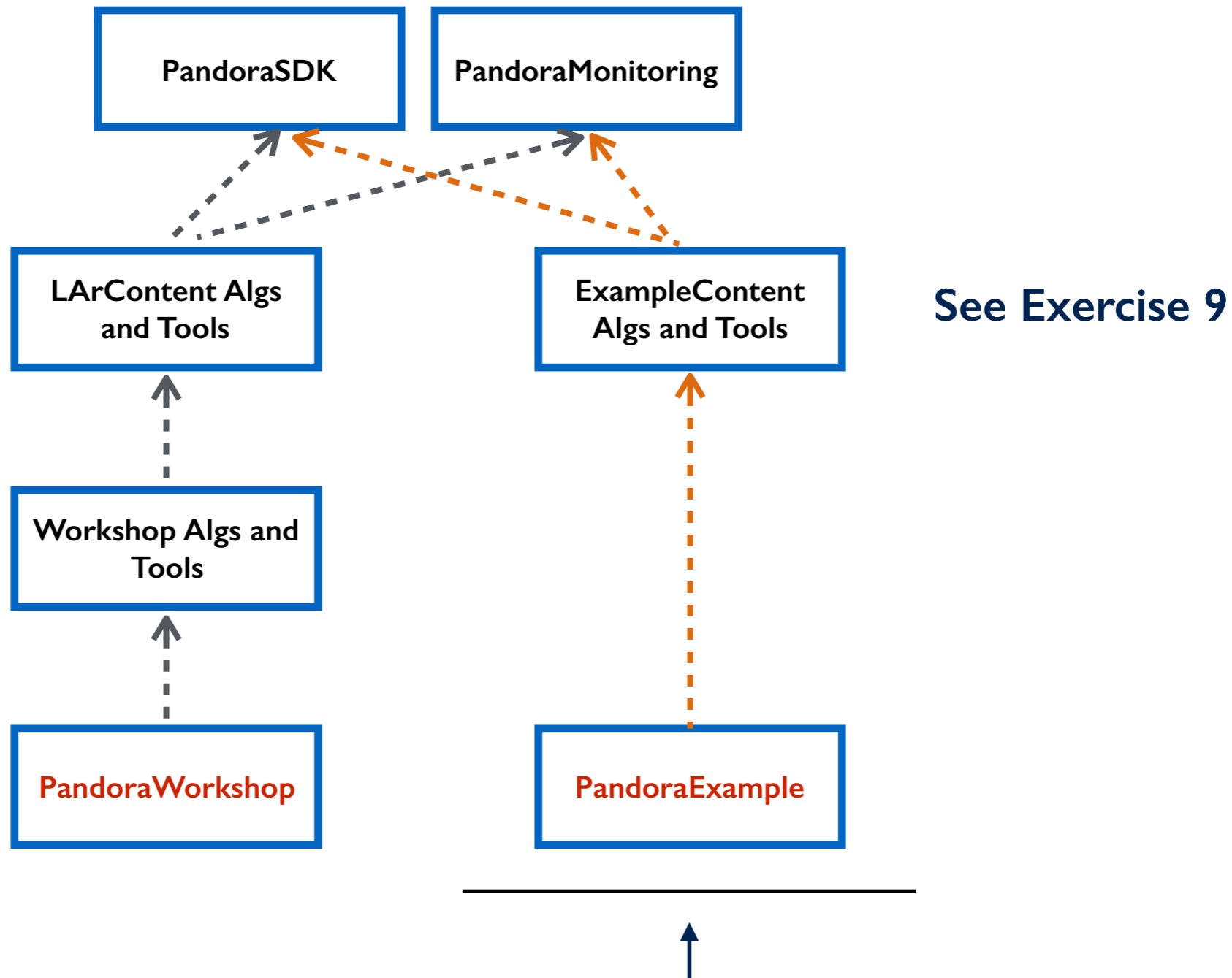


Main workshop development area, to which we will add new LAr-specific Algorithms and Tools

All LAr TPC Algorithms, Tools and Helper functions are accessible/reusable



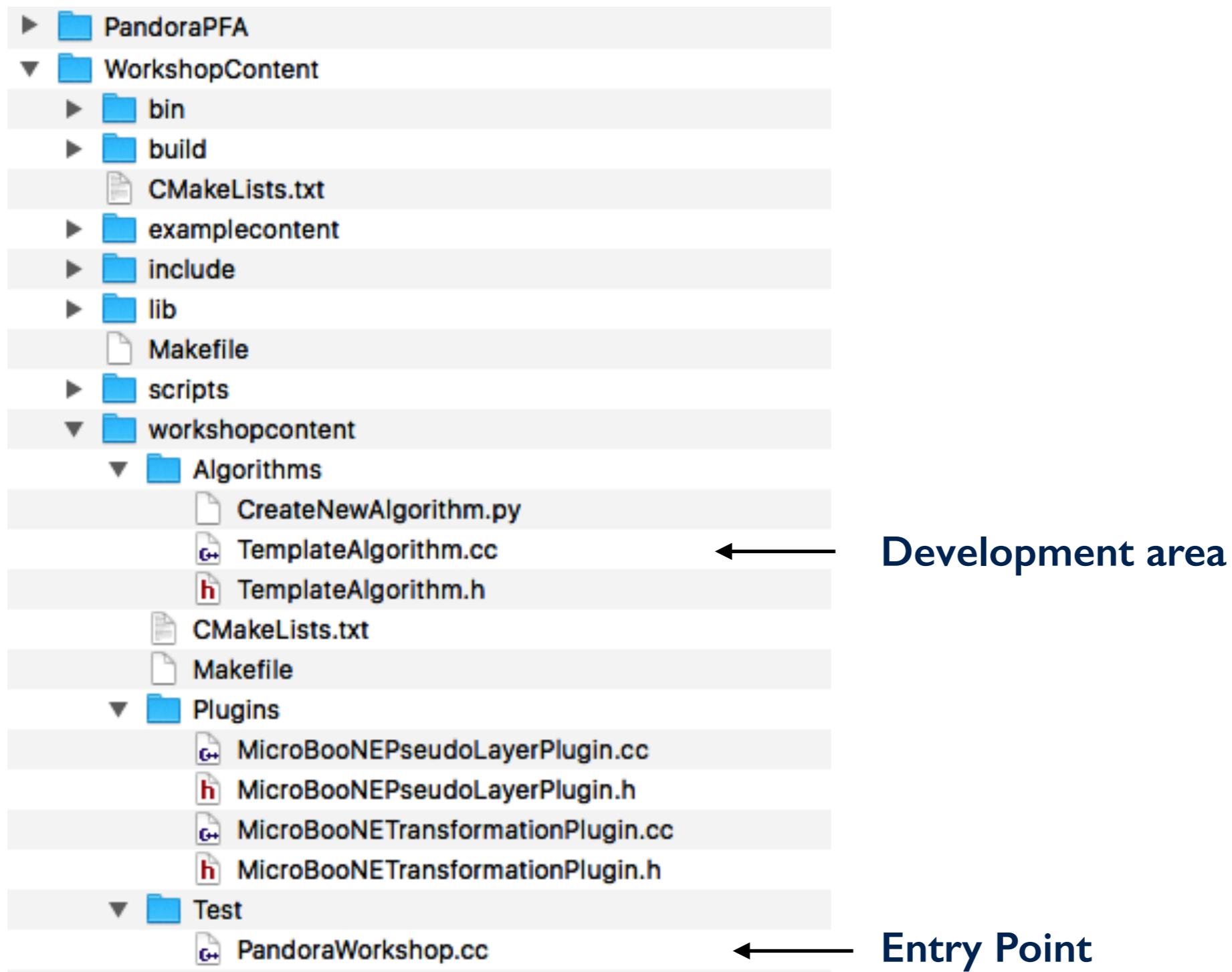
Workshop Test Area



Completely generic Pandora learning/test library, demonstrating many algorithm types with placeholder logic. Will look at this as required throughout workshop.



Directory Structure



Development area

Entry Point



Adding a New Algorithm

```
cd $MY_TEST_AREA/WorkshopContent/workshopcontent/Algorithms # Should see "TemplateAlgorithm" present here  
python CreateNewAlgorithm.py --name MyTest
```

New Algorithm name: MyTestAlgorithm

Screen Output

```
--in workshopcontent/Test/PandoraWorkshop.cc  
  
#include "workshopcontent/Algorithms/MyTestAlgorithm.h"  
  
PANDORA_THROW_RESULT_IF(pandora::STATUS_CODE_SUCCESS, !=, PandoraApi::RegisterAlgorithmFactory(*pPandora, "MyTest",  
new workshop_content::MyTestAlgorithm::Factory));  
  
--in scripts/PandoraSettings_Workshop.xml  
  
<algorithm type = "MyTest"/>
```



Steps required to register and run algorithm with Pandora



Algorithm Implementation



```
/**
 * @file   WorkshopContent/workshopcontent/Algorithms/MyTestAlgorithm.cc
 *
 * @brief  Implementation of the mytest algorithm class.
 *
 * $Log: $
 */

#include "Pandora/AlgorithmHeaders.h"

#include "workshopcontent/Algorithms/MyTestAlgorithm.h"

using namespace pandora;

namespace workshop_content
{
  StatusCode MyTestAlgorithm::Run()
  {
    // Algorithm code here

    return STATUS_CODE_SUCCESS;
  }

  //-----

  StatusCode MyTestAlgorithm::ReadSettings(const TiXmlHandle /*xmlHandle*/)
  {
    // Read settings from xml file here

    return STATUS_CODE_SUCCESS;
  }
} // namespace workshop_content
```

A really basic starting point, with just Run and ReadSettings Callbacks



Algorithm Declaration



```
/**
 * @file WorkshopContent/workshopcontent/Algorithms/MyTestAlgorithm.h
 *
 * @brief Header file for the mytest algorithm class.
 *
 * $Log: $
 */
#ifndef WORKSHOP_MYTEST_ALGORITHM_H
#define WORKSHOP_MYTEST_ALGORITHM_H 1

#include "Pandora/Algorithm.h"

namespace workshop_content
{
    /**
     * @brief MyTestAlgorithm class
     */
    class MyTestAlgorithm : public pandora::Algorithm
    {
    public:
        /**
         * @brief Factory class for instantiating algorithm
         */
        class Factory : public pandora::AlgorithmFactory
        {
        public:
            pandora::Algorithm *CreateAlgorithm() const;
        };

    private:
        pandora::StatusCode Run();
        pandora::StatusCode ReadSettings(const pandora::TiXmlHandle xmlHandle);

        // Member variables here
    };

    //-----

    inline pandora::Algorithm *MyTestAlgorithm::Factory::CreateAlgorithm() const
    {
        return new MyTestAlgorithm();
    }

} // namespace workshop_content

#endif // #ifndef WORKSHOP_MYTEST_ALGORITHM_H
```

A really basic starting point
Note: **AlgorithmFactory**



Registering the Algorithm



```
#include "workshopcontent/Algorithms/MyTestAlgorithm.h"
```



Pick-up AlgorithmFactory
declaration and implementation

```
...
```

```
int main(int argc, char *argv[])  
{
```

```
  try  
  {
```

```
    Parameters parameters;
```

```
    if (!ParseCommandLine(argc, argv, parameters))  
      return 1;
```

```
#ifdef MONITORING
```

```
  TApplication *const pTApplication = new TApplication("Workshop", &argc, argv);  
  pTApplication->SetReturnFromRun(kTRUE);
```

```
#endif
```

```
  const pandora::Pandora *const pPandora = new pandora::Pandora();
```

```
  PANDORA_THROW_RESULT_IF(pandora::STATUS_CODE_SUCCESS, !=, LArContent::RegisterAlgorithms(*pPandora));
```

```
  PANDORA_THROW_RESULT_IF(pandora::STATUS_CODE_SUCCESS, !=, LArContent::RegisterBasicPlugins(*pPandora));
```

```
  PANDORA_THROW_RESULT_IF(pandora::STATUS_CODE_SUCCESS, !=, LArContent::SetLArPseudoLayerPlugin(*pPandora,  
    new workshop_content::MicroBooNEPseudoLayerPlugin));
```

```
  PANDORA_THROW_RESULT_IF(pandora::STATUS_CODE_SUCCESS, !=, LArContent::SetLArTransformationPlugin(*pPandora,  
    new workshop_content::MicroBooNETransformationPlugin));
```

```
  PANDORA_THROW_RESULT_IF(pandora::STATUS_CODE_SUCCESS, !=, PandoraApi::RegisterAlgorithmFactory(*pPandora,  
    "MyTest", new workshop_content::MyTestAlgorithm::Factory));
```



Give a
Factory
instance to
Pandora

```
  PANDORA_THROW_RESULT_IF(pandora::STATUS_CODE_SUCCESS, !=, PandoraApi::ReadSettings(*pPandora,  
    parameters.m_pandoraSettingsFile));
```

```
  int nEvents(0);
```

```
  while ((nEvents++ < parameters.m_nEventsToProcess) || (0 > parameters.m_nEventsToProcess))
```

```
  {
```

```
    PANDORA_THROW_RESULT_IF(pandora::STATUS_CODE_SUCCESS, !=, PandoraApi::ProcessEvent(*pPandora));
```

```
    PANDORA_THROW_RESULT_IF(pandora::STATUS_CODE_SUCCESS, !=, PandoraApi::Reset(*pPandora));
```

```
  }
```

```
  delete pPandora;
```

```
}
```

```
catch (pandora::StatusCodeException &statusCodeException)
```

```
{
```

```
  std::cerr << "Pandora Exception caught: " << statusCodeException.ToString() << std::endl;
```

```
  return 1;
```

```
}
```

```
  return 0;
```

```
}
```

[\\$MY_TEST_AREA/WorkshopContent/workshopcontent/Test/PandoraWorkshop.cc](#)



Building the New Algorithm

```
cd $MY_TEST_AREA/WorkshopContent/build
# Don't forget you'll need to re-run CMake after adding a new source file
make install
```

```
bash-3.2$ cd $MY_TEST_AREA/WorkshopContent/build/
bash-3.2$ make install
Scanning dependencies of target WorkshopContent
[ 3%] Building CXX object workshopcontent/CMakeFiles/WorkshopContent.dir/Algorithms/MyTestAlgorithm.cc.o
[ 6%] Linking CXX shared library lib/libWorkshopContent.dylib
[ 16%] Built target WorkshopContent
Scanning dependencies of target PandoraWorkshop
[ 20%] Building CXX object workshopcontent/CMakeFiles/PandoraWorkshop.dir/Test/PandoraWorkshop.cc.o
[ 23%] Linking CXX executable bin/PandoraWorkshop
[ 23%] Built target PandoraWorkshop
[ 93%] Built target ExampleContent
[100%] Built target PandoraExample
Install the project...
-- Install configuration: "RelWithDebInfo"
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Algorithms
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Algorithms/MyTestAlgorithm.h
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Algorithms/TemplateAlgorithm.h
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Plugins
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Plugins/MicroBooNEPseudoLayerPlugin.h
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Plugins/MicroBooNETransformationPlugin.h
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/workshopcontent/Test
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/examplecontent
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/examplecontent/ExampleAlgorithms
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/examplecontent/ExampleAlgorithms/AccessListsAlgorithm.h
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/examplecontent/ExampleAlgorithms/CreateAdditionalCaloHitsAlgorithm.h
-- Up-to-date: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/include/examplecontent/ExampleAlgorithms/CreateClustersAlgorithm.h
```



Running the New Algorithm



```

<pandora>
  <!-- GLOBAL SETTINGS -->
  <IsMonitoringEnabled>true</IsMonitoringEnabled>
  <ShouldDisplayAlgorithmInfo>true</ShouldDisplayAlgorithmInfo>
  <SingleHitTypeClusteringMode>true</SingleHitTypeClusteringMode>

  <!-- ALGORITHM SETTINGS -->
  <algorithm type = "LArEventReading">
    <EventFileName>/path/to/Events_MicroBooNE.pndr</EventFileName>
    <GeometryFileName>/path/to/Geometry_MicroBooNE.pndr</GeometryFileName>
    <ShouldReadEvents>true</ShouldReadEvents>
    <ShouldReadGeometry>true</ShouldReadGeometry>
    <SkipToEvent>0</SkipToEvent>
  </algorithm>

  <!-- LAR TPC EVENT RECONSTRUCTION -->
  <algorithm type = "LArListPreparation">
    <OnlyAvailableCaloHits>true</OnlyAvailableCaloHits>
    <OutputCaloHitListNameW>CaloHitListW</OutputCaloHitListNameW>
    <OutputCaloHitListNameU>CaloHitListU</OutputCaloHitListNameU>
    <OutputCaloHitListNameV>CaloHitListV</OutputCaloHitListNameV>
    <FilteredCaloHitListName>CaloHitList2D</FilteredCaloHitListName>
    <CurrentCaloHitListReplacement>CaloHitListW</CurrentCaloHitListReplacement>
    <OutputMCParticleListNameU>MCParticleListU</OutputMCParticleListNameU>
    <OutputMCParticleListNameV>MCParticleListV</OutputMCParticleListNameV>
    <OutputMCParticleListNameW>MCParticleListW</OutputMCParticleListNameW>
    <OutputMCParticleListName3D>MCParticleList3D</OutputMCParticleListName3D>
    <CurrentMCParticleListReplacement>MCParticleList3D</CurrentMCParticleListReplacement>
    <MipEquivalentCut>0.</MipEquivalentCut>
  </algorithm>

  <algorithm type = "MyTest"/>

  <algorithm type = "LArVisualMonitoring">
    <CaloHitListNames>CaloHitListW CaloHitListU CaloHitListV</CaloHitListNames>
    <MCParticleListNames>MCParticleList3D</MCParticleListNames>
    <SuppressMCParticles>22:0.01 2112:1.0</SuppressMCParticles>
    <ShowDetector>true</ShowDetector>
  </algorithm>
</pandora>

```

← Where you specify input events and geometry files

Event samples available [here](#) (docdb username/password)

← New algorithm, with (as yet) no configuration details req'd

`$MY_TEST_AREA/WorkshopContent/scripts/PandoraSettings_Workshop.xml`



Running the New Algorithm

```
$MY_TEST_AREA/WorkshopContent/bin/PandoraWorkshop -?
```

PandoraWorkshop

```
-i PandoraSettings.xml (mandatory)
-n NEventsToProcess    (mandatory)
-N                      (optional, display event numbers)
```

```
$MY_TEST_AREA/WorkshopContent/bin/PandoraWorkshop \
-i $MY_TEST_AREA/WorkshopContent/scripts/PandoraSettings_Workshop.xml \
-n 10
```

```
Failure in reading pandora settings, STATUS_CODE_FAILURE
PandoraApi::ReadSettings(*pPandora, parameters.m_pandoraSettingsFile) throw STATUS_CODE_FAILURE
in function: main
in file: /Users/johnmarshall/Documents/WorkshopTest/WorkshopContent/workshopcontent/Test/
PandoraWorkshop.cc line#: 80
Pandora Exception caught: STATUS_CODE_FAILURE
```



Need to point to valid event and geometry files, available [here](#) (docdb username/password)



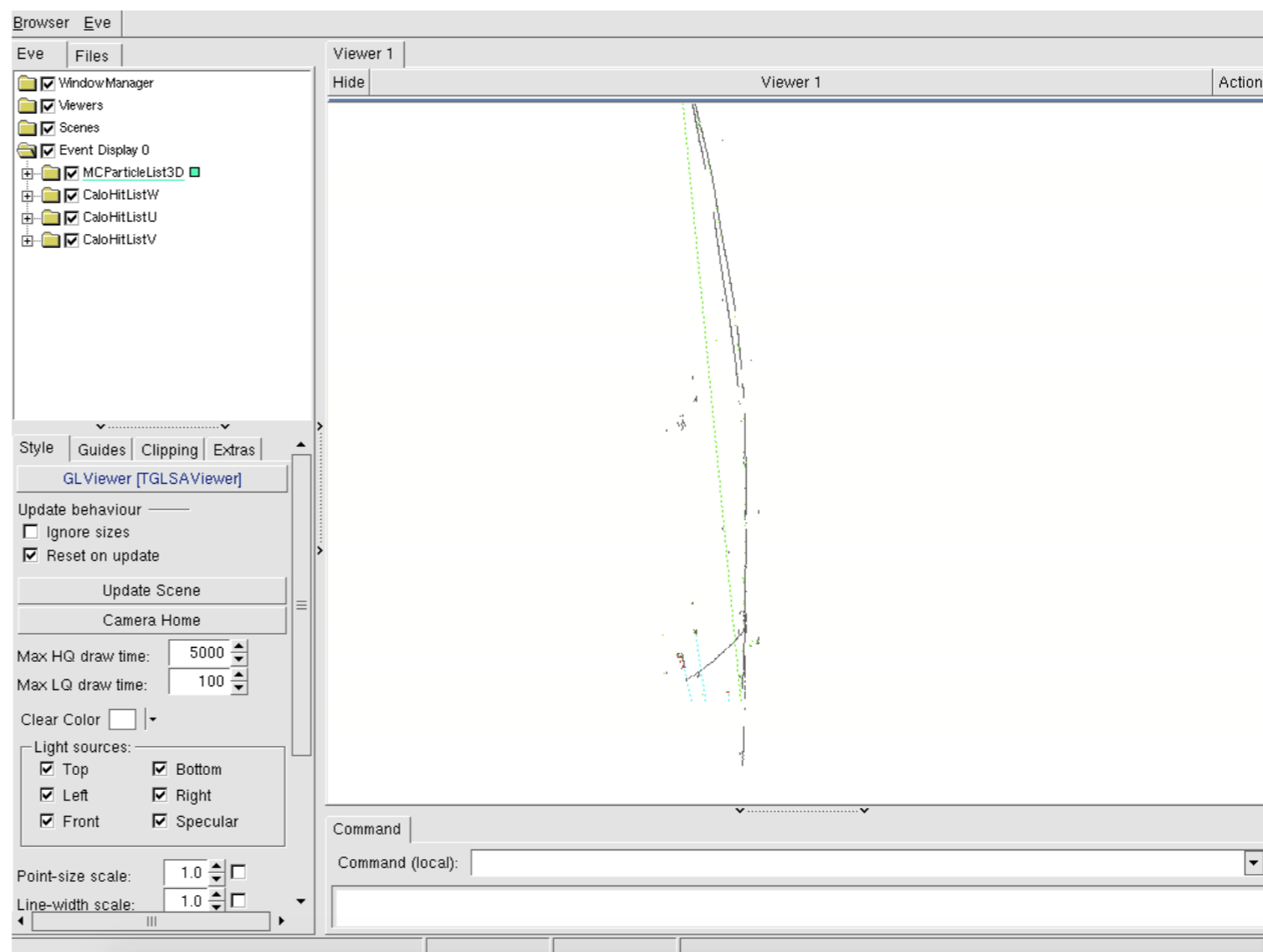
Running the Algorithm



```
$MY_TEST_AREA/WorkshopContent/bin/PandoraWorkshop \
-i $MY_TEST_AREA/WorkshopContent/scripts/PandoraSettings_Workshop.xml \
-n 10
```

PROCESSING EVENT: 0

```
> Running Algorithm: 0x7f93915b63c0, LArEventReading
> Running Algorithm: 0x7f939fab5250, LArListPreparation
> Running Algorithm: 0x7f939beea8b0, MyTest
> Running Algorithm: 0x7f939beec180, LArVisualMonitoring
PandoraMonitoring::InitializeEve(): DISPLAY environment set to /path/org.macosforge.xquartz:0
Info in <TGeoManager::TGeoManager>: Geometry DetectorGeometry, detector geometry created
Info in <TGeoManager::SetTopVolume>: Top volume is Detector. Master volume is Detector
Info in <TGeoNavigator::BuildCache>: --- Maximum geometry depth set to 100
Info in <TGeoManager::SetVisLevel>: Automatic visible depth disabled
Press return to continue ...
```





Next Exercise: Add Algorithm Implementation for Clustering