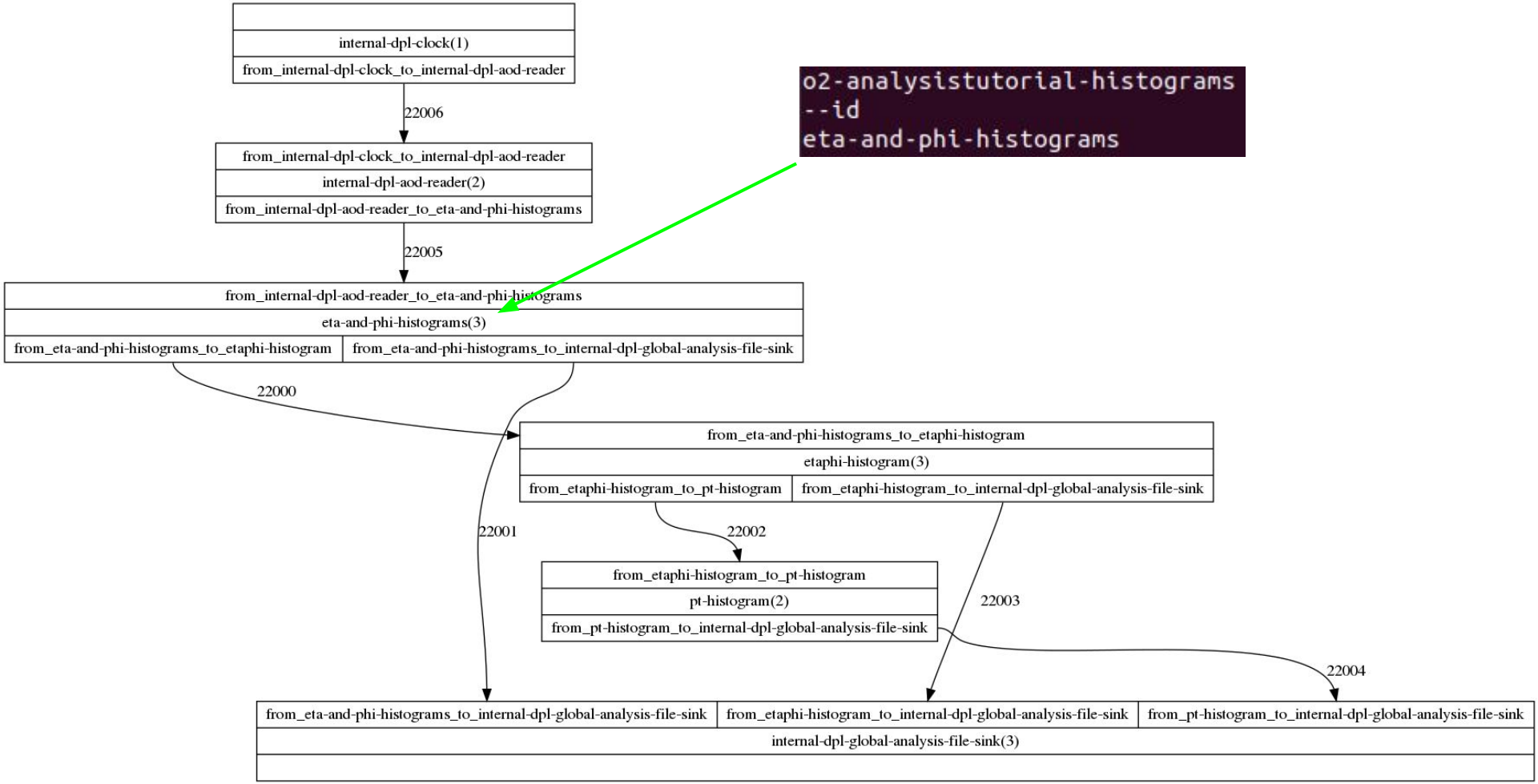
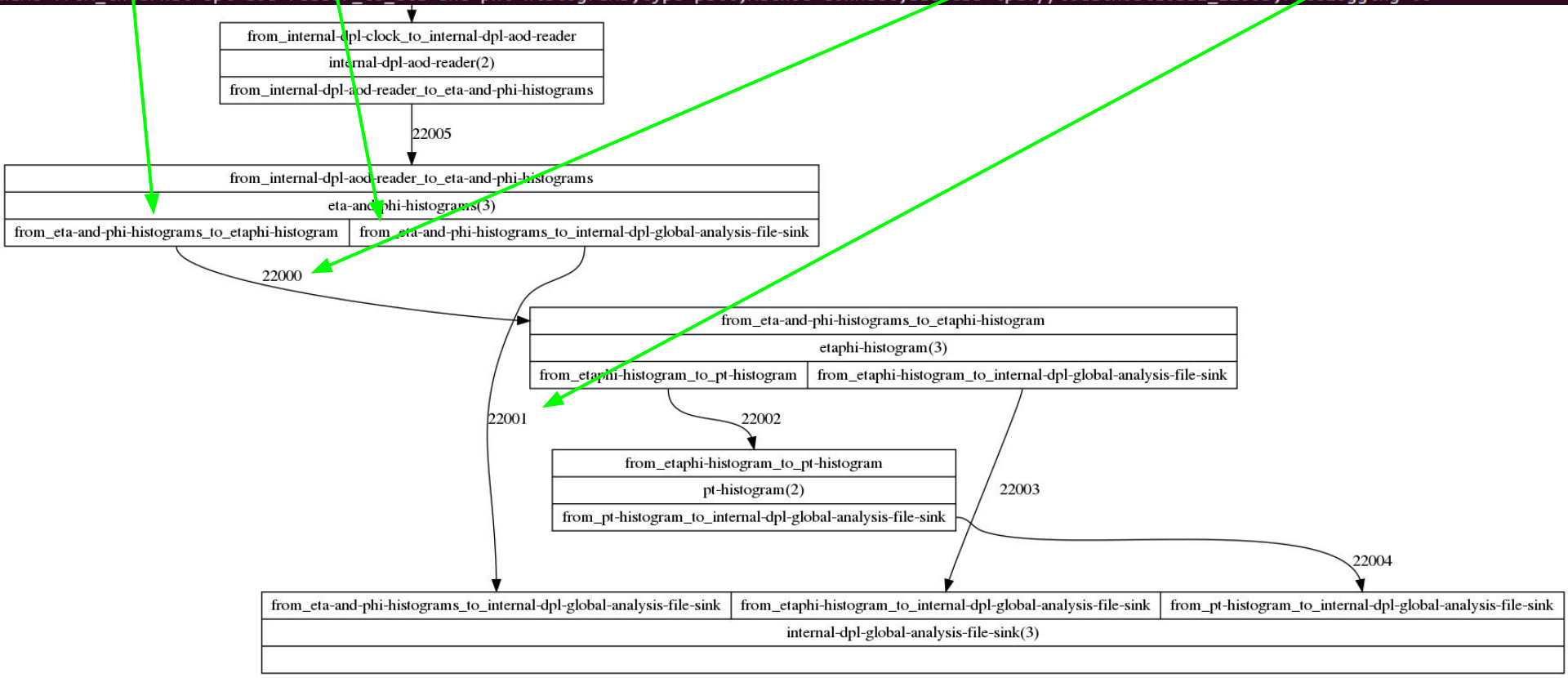
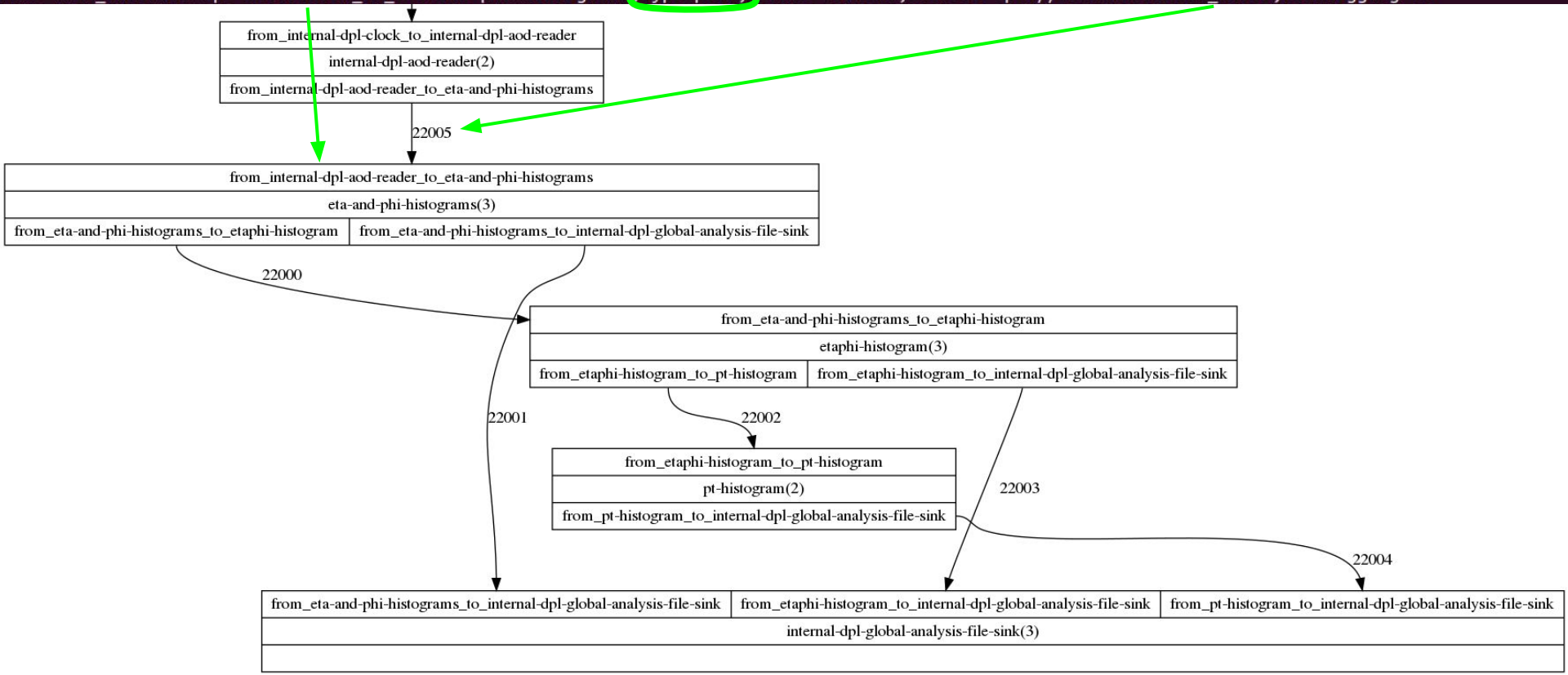

internal-dpl-clock(1)
from_internal-dpl-clock_to_internal-dpl-aod-reader
22006
from_internal-dpl-clock_to_internal-dpl-aod-reader
internal-dpl-aod-reader(2)
from_internal-dpl-aod-reader_to_eta-and-phi-histograms
22005
from_internal-dpl-aod-reader_to_eta-and-phi-histograms
eta-and-phi-histograms(3)
from_eta-and-phi-histograms_to_etaphi-histogram
from_eta-and-phi-histograms_to_internal-dpl-global-analysis-file-sink
o2-analysistutorial-histograms
--id
eta-and-phi-histograms
22000
from_eta-and-phi-histograms_to_etaphi-histogram
etaphi-histogram(3)
from_etaphi-histogram_to_pt-histogram
from_etaphi-histogram_to_internal-dpl-global-analysis-file-sink
22001
22002
22003
from_etaphi-histogram_to_pt-histogram
pt-histogram(2)
from_pt-histogram_to_internal-dpl-global-analysis-file-sink
22004
from_eta-and-phi-histograms_to_internal-dpl-global-analysis-file-sink
from_etaphi-histogram_to_internal-dpl-global-analysis-file-sink
from_pt-histogram_to_internal-dpl-global-analysis-file-sink
internal-dpl-global-analysis-file-sink(3)

```
--channel-config
name=from_eta-and-phi-histograms_to_etaphi-histogram,type=push,method=bind,address=ipc://localhost16332_22000,rateLogging=60
--channel-config
name=from_eta-and-phi-histograms_to_internal-dpl-global-analysis-file-sink,type=push,method=bind,address=ipc://localhost16332_22001,rateLogging=60
--channel-config
name=from_internal-dpl-aod-reader_to_eta-and-phi-histograms,type=pull,method=connect,address=ipc://localhost16332_22005,rateLogging=60
```



```
--channel-config
name=from_eta-and-phi-histograms_to_etaphi-histogram,type=push,method=bind,address=ipc://localhost16332_22000,rateLogging=60
--channel-config
name=from_eta-and-phi-histograms_to_internal-dpl-global-analysis-file-sink,type=push,method=bind,address=ipc://localhost16332_22001,rateLogging=60
--channel-config
name=from_internal-dpl-aod-reader_to_eta-and-phi-histograms,type=pull,method=connect,address=ipc://localhost16332_22005,rateLogging=60
```



```
[7126:eta-and-phi-histograms]: ==> Dumping object at: 0x000055ba38f43450, name=phi, class=TH1F
[7126:eta-and-phi-histograms]:
[7126:eta-and-phi-histograms]: fNcells                      102              number of bins(1D), cells (2D) +U/Overflows
[7126:eta-and-phi-histograms]: fXaxis                       ->55ba38f434c8   X axis descriptor
[7126:eta-and-phi-histograms]: fXaxis.fNbins                100              Number of bins
[7126:eta-and-phi-histograms]: fXaxis.fXmin                 0                low edge of first bin
[7126:eta-and-phi-histograms]: fXaxis.fXmax                 6.28319          upper edge of last bin
[7126:eta-and-phi-histograms]: fXaxis.fXbins                ->55ba38f43548   Bin edges array in X
[7126:eta-and-phi-histograms]: fXaxis.fXbins.*fArray        ->0              [fN] Array of fN doubles
[7126:eta-and-phi-histograms]: fXaxis.fXbins.fN             0                Number of array elements
[7126:eta-and-phi-histograms]: fXaxis.fFirst                0                first bin to display
[7126:eta-and-phi-histograms]: fXaxis.fLast                 0                last bin to display
[7126:eta-and-phi-histograms]: fXaxis.fBits2                0                second bit status word
[7126:eta-and-phi-histograms]: fXaxis.fTimeDisplay          false            on/off displaying time values instead of numerics
[7126:eta-and-phi-histograms]: fXaxis.fTimeFormat                            Date&time format, ex: 09/12/99 12:34:00
[7126:eta-and-phi-histograms]: fXaxis.fTimeFormat.fRep      ->55ba38f43578   ! String data
[7126:eta-and-phi-histograms]: fXaxis.fTimeFormat.fRep.     ->55ba38f43578
[7126:eta-and-phi-histograms]: fXaxis.*fParent              ->55ba38f43450   !Object owning this axis
[7126:eta-and-phi-histograms]: fXaxis.*fLabels              ->0              List of labels
[7126:eta-and-phi-histograms]: fXaxis.*fModLabs             ->0              List of modified labels
[7126:eta-and-phi-histograms]: fXaxis.fName                 xaxis            object identifier
[7126:eta-and-phi-histograms]: fXaxis.fName.fRep            ->55ba38f434e0   ! String data
[7126:eta-and-phi-histograms]: fXaxis.fName.fRep.           ->55ba38f434e0
[7126:eta-and-phi-histograms]: fXaxis.fTitle                                 object title
[7126:eta-and-phi-histograms]: fXaxis.fTitle.fRep           ->55ba38f434f8   ! String data
[7126:eta-and-phi-histograms]: fXaxis.fTitle.fRep.          ->55ba38f434f8
[7126:eta-and-phi-histograms]: fXaxis.fUniqueID             0                object unique identifier
[7126:eta-and-phi-histograms]: fXaxis.fBits                 0x02000000       bit field status word
[7126:eta-and-phi-histograms]: fXaxis.fNdivisions           510              Number of divisions(10000*n3 + 100*n2 + n1)
[7126:eta-and-phi-histograms]: fXaxis.fAxisColor            1                Color of the line axis
[7126:eta-and-phi-histograms]: fXaxis.fLabelColor           1                Color of labels
[7126:eta-and-phi-histograms]: fXaxis.fLabelFont            42               Font for labels
[7126:eta-and-phi-histograms]: fXaxis.fLabelOffset          0.005            Offset of labels
[7126:eta-and-phi-histograms]: fXaxis.fLabelSize            0.035            Size of labels
[7126:eta-and-phi-histograms]: fXaxis.fTickLength           0.03             Length of tick marks
[7126:eta-and-phi-histograms]: fXaxis.fTitleOffset          1                Offset of axis title
[7126:eta-and-phi-histograms]: fXaxis.fTitleSize            0.035            Size of axis title
[7126:eta-and-phi-histograms]: fXaxis.fTitleColor           1                Color of axis title
[7126:eta-and-phi-histograms]: fXaxis.fTitleFont            42               Font for axis title
[7126:eta-and-phi-histograms]: fYaxis                       ->55ba38f435a0   Y axis descriptor
[7126:eta-and-phi-histograms]: fYaxis.fNbins                1                Number of bins
[7126:eta-and-phi-histograms]: fYaxis.fXmin                 0                low edge of first bin
[7126:eta-and-phi-histograms]: fYaxis.fXmax                 1                upper edge of last bin
[7126:eta-and-phi-histograms]: fYaxis.fXbins                ->55ba38f43620   Bin edges array in X
[7126:eta-and-phi-histograms]: fYaxis.fXbins.*fArray        ->0              [fN] Array of fN doubles
```

```
/// Send a snapshot of an object, depending on the object type it is serialized before.
/// The method always takes a copy of the data, which will then be sent once the
/// computation ends.
/// Framework does not take ownership of the @a object. Changes to @a object
/// after the call will not be sent.
///
/// Supported types:
/// - messageable types (trivially copyable, non-polymorphic
/// - std::vector of messageable types
/// - std::vector of pointers of messageable type
/// - types with ROOT dictionary and implementing the ROOT ClassDef interface
///
/// Note: for many use cases, especially for the messageable types, the `make` interface
/// might be better suited as the objects are allocated directly in the underlying
/// memory resource and the copy can be avoided.
///
/// Note: messageable objects with ROOT dictionary are preferably sent unserialized.
/// Use @a ROOTSerialized type wrapper to force ROOT serialization. Same applies to
/// types which do not implement the ClassDef interface but have a dictionary.
template <typename T>
void snapshot(const Output& spec, T const& object)
{
  /* Reszta kodu funkcji */
  if constexpr (std::is_base_of_v<TObject, T>)
  {
      if (DATA_DUMP)
      {
        std::cout << "===DUMP===" << std::endl;
        object.Dump();
      }
  }
  addPartToContext(std::move(payloadMessage), spec, serializationType);
}
```

O2/Framework/Core/include/Framework/DataAllocator.h

```
dak@dak:~/alice$ grep -r snapshot O2/Framework/Core/src/*
O2/Framework/Core/src/DataAllocator.cxx:void DataAllocator::snapshot(const Output& spec, const char* payload, size_t payloadSize,
O2/Framework/Core/src/Dispatcher.cxx:  dataAllocator.snapshot(output, inputData.payload, inputHeader->payloadSize, inputHeader->payloadSerializationMethod);
dak@dak:~/alice$ grep -r snapshot O2/Framework/Core/include/Framework/*
O2/Framework/Core/include/Framework/AnalysisTask.h:    context.outputs().snapshot(what.ref(), *what);
O2/Framework/Core/include/Framework/DataAllocator.h:  /// Send a snapshot of an object, depending on the object type it is serialized before.
O2/Framework/Core/include/Framework/DataAllocator.h:  void snapshot(con
O2/Framework/Core/include/Framework/DataAllocator.h:    // Serialize
O2/Framework/Core/include/Framework/DataAllocator.h:      // Seriali
O2/Framework/Core/include/Framework/DataAllocator.h:      return snap
O2/Framework/Core/include/Framework/DataAllocator.h:    // Serialize
O2/Framework/Core/include/Framework/DataAllocator.h:      // Explici
O2/Framework/Core/include/Framework/DataAllocator.h:  /// Take a snaps
O2/Framework/Core/include/Framework/DataAllocator.h:  void snapshot(con
O2/Framework/Core/include/Framework/DataAllocator.h:  /// snapshot obje
O2/Framework/Core/include/Framework/DataAllocator.h:  auto snapshot(Ou
O2/Framework/Core/include/Framework/DataAllocator.h:    return snapsho
O2/Framework/Core/include/Framework/SerializationMethods.h:///   output
O2/Framework/Core/include/Framework/SerializationMethods.h:///   output
O2/Framework/Core/include/Framework/SerializationMethods.h:///   output
O2/Framework/Core/include/Framework/SerializationMethods.h:///   output
```

```cpp
template <typename T>
struct OutputManager<OutputObj<T>> {
  static bool appendOutput(std::vector<OutputSpec>& outputs, OutputObj<T>& what, uint32_t hash)
  {
    what.setHash(hash);
    outputs.emplace_back(what.spec());
    return true;
  }
  static bool prepare(ProcessingContext&, OutputObj<T>&)
  {
    return true;
  }

  static bool finalize(ProcessingContext&, OutputObj<T>&)
  {
    return true;
  }

  static bool postRun(EndOfStreamContext& context, OutputObj<T>& what)
  {
    context.outputs().snapshot(what.ref(), *what);
    return true;
  }
};
```

```cpp
template <typename T, typename... Args>
DataProcessorSpec adaptAnalysisTask(char const* name, Args&&... args)
{
  TH1::AddDirectory(false);
  auto task = std::make_shared<T>(std::forward<Args>(args)...);
  auto hash = compile_time_hash(name);

  std::vector<OutputSpec> outputs;
  std::vector<ConfigParamSpec> options;

  auto tupledTask = o2::framework::to_tuple_refs(*task.get());
  static_assert(has_process<T>::value || has_run<T>::value || has_init<T>::value,
                "At least one of process(...), T::run(...), init(...) must be defined");

  std::vector<InputSpec> inputs;
  std::vector<ExpressionInfo> expressionInfos;

  if constexpr (has_process<T>::value) {
    // this pushes (I,schemaPtr,nullptr) into expressionInfos for arguments that are Filtered/filtered_iterators
    AnalysisDataProcessorBuilder::inputsFromArgs(&T::process, inputs, expressionInfos);
    // here the FilterManager will prepare the gandiva trees matched to schemas and put the pointers into expressionInfos
    std::apply([&expressionInfos](auto&... x) {
      return (FilterManager<std::decay_t<decltype(x)>>::createExpressionTrees(x, expressionInfos), ...);
    },
               tupledTask);
  }

  std::apply([&outputs, &hash](auto&... x) { return (OutputManager<std::decay_t<decltype(x)>>::appendOutput(outputs, x, hash), ...
  std::apply([&options, &hash](auto&... x) { return (OptionManager<std::decay_t<decltype(x)>>::appendOption(options, x), ...); }

  auto algo = AlgorithmSpec::InitCallback{[task, expressionInfos](InitContext& ic) {
    auto tupledTask = o2::framework::to_tuple_refs(*task.get());
    std::apply([&ic](auto&&... x) { return (OptionManager<std::decay_t<decltype(x)>>::prepare(ic, x), ...); }, tupledTask);

    auto& callbacks = ic.services().get<CallbackService>();
    auto endofdatacb = [task](EndOfStreamContext& eosContext) {
      auto tupledTask = o2::framework::to_tuple_refs(*task.get());
      std::apply([&eosContext](auto&&... x) { return (OutputManager<std::decay_t<decltype(x)>>::postRun(eosContext, x), ...); },
      eosContext.services().get<ControlService>().readyToQuit(QuitRequest::Me);
    };
```