DABC version 2.0

Sergey Linev,

EE meeting,

22.04.2013

DABC - history

- 2005 FutureDAQ project
- 2007 first prototype, using a lot of XDAQ
- 2008 first CBM beamtime, remove XDAQ
- 2009 version 1.0
- 2011 intermediate 1.9 version
- 2013 “final” 2.0 version

- used in 9 CBM beam tests in GSI/FZJ/CERN

Why we need DABC?

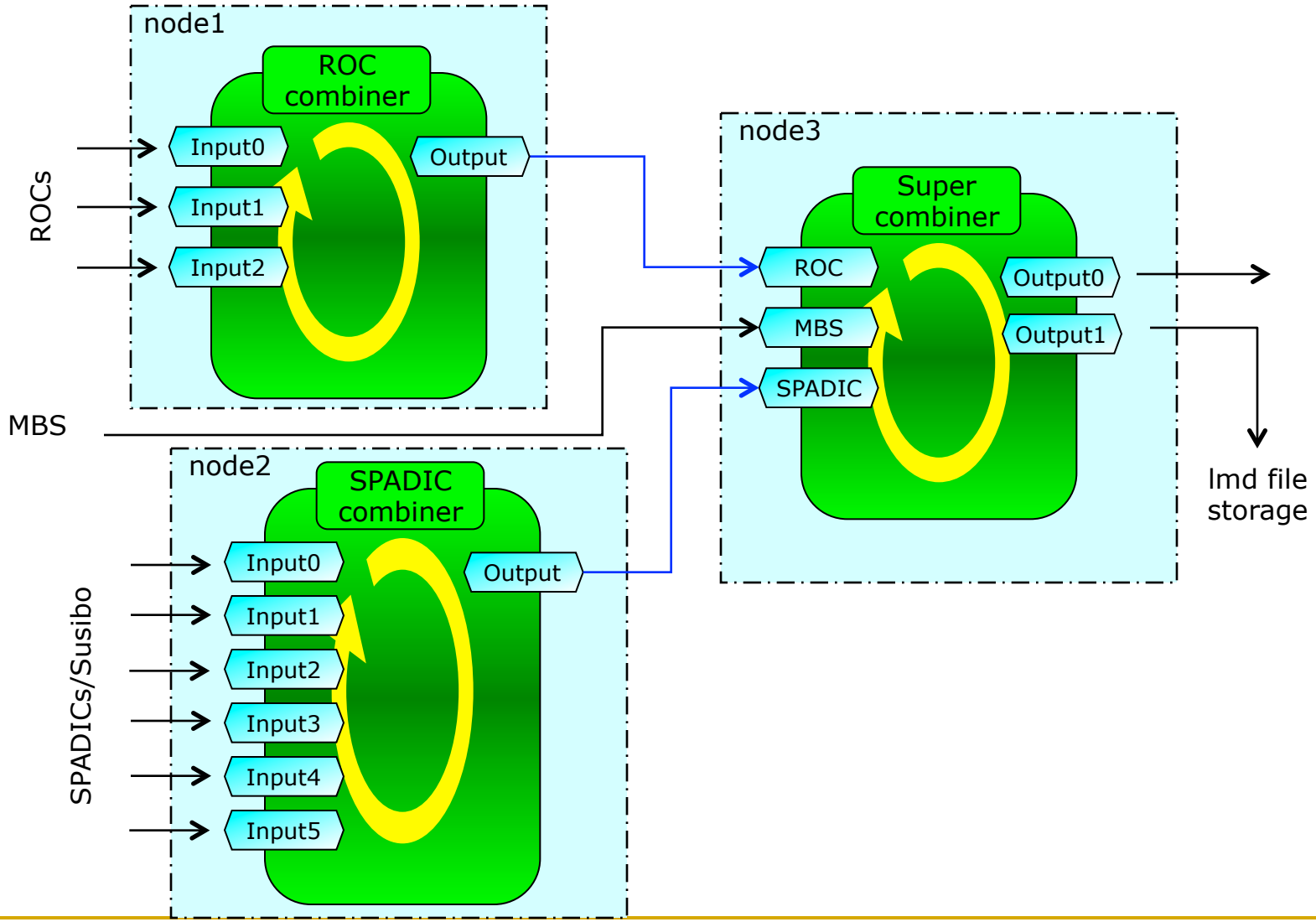
- Connect (near) any frontend
- Handle (together) triggered and self-triggered data
- Merge / transform / distribute data-streams over many computing nodes
- Provide interfaces for application code

DABC - main features

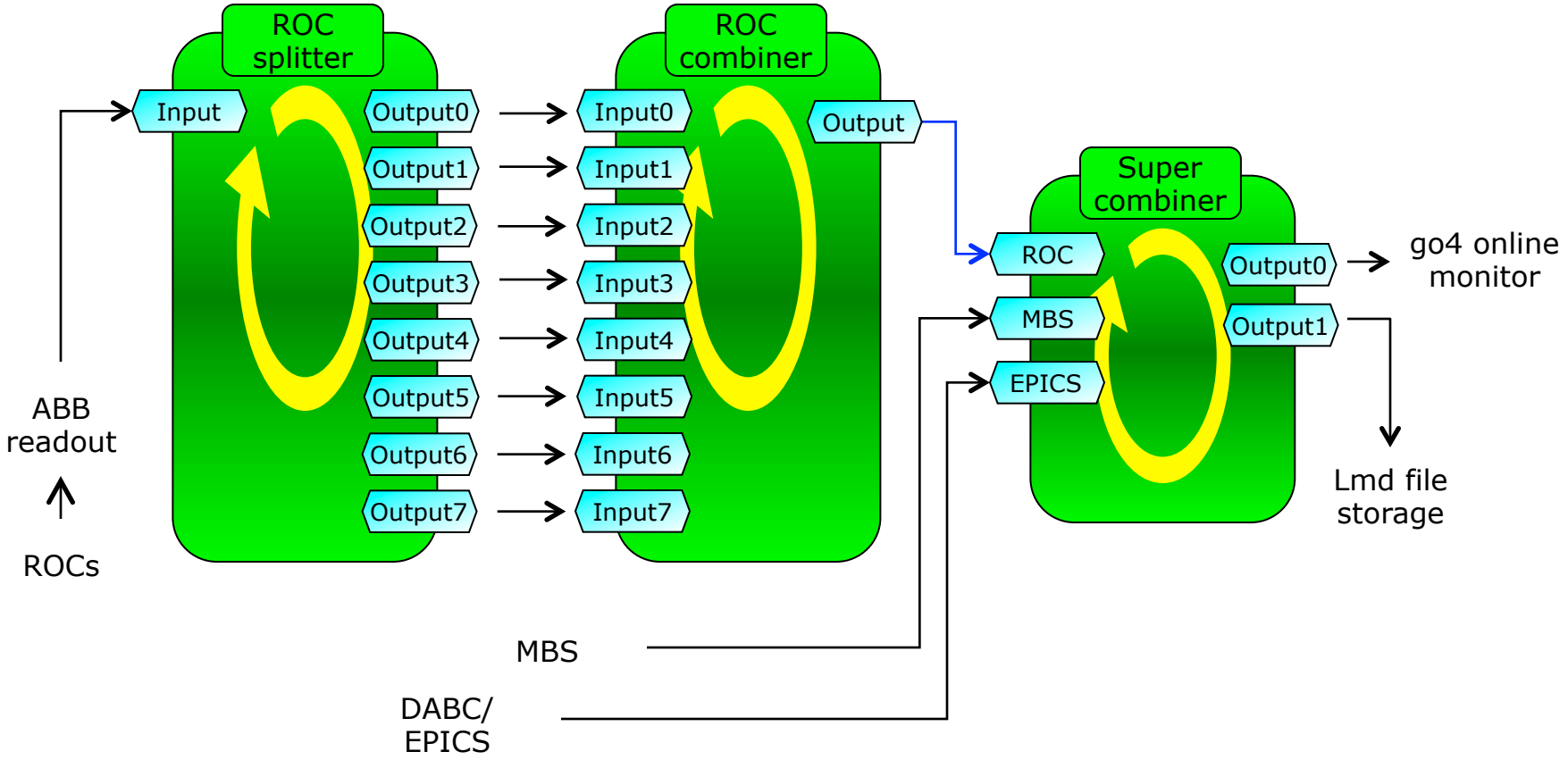
- Compact multi-threaded zero-copy data-flow core

- Number of device/application specific add-ons
- TCP/IP (sockets) and InfiniBand (OFED verbs) as data transports
- Plugins for user-specific components
- BNET – components for constructing event-building network
- Flexible configurations with xml files
- DIM as interface for control system
- generic Java GUI

DABC in CBM test beams



DABC in CBM test beams



DABC 1.0 - open issues

- In some situations API is not thread-safe
  - object deletion in concurrent thread
  - no automatic cleanup in case of exceptions
  - regular shutdown is most complex procedure in the framework
  - as result, user should be aware about all these problems
- Complex API for transport implementations
- Master-slave topology
  - clients cannot communicate directly with each other
  - master is single point of failure
  - scalability doubts from 100 nodes
- Control DIM interface is too strongly coupled with core DABC functionality

DABC 2.0 – new solutions

- Reference (smart pointer) class
- Flexible multi-node handling
- Separate input and output ports

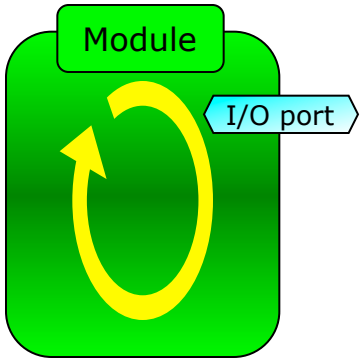
- New transport interface

DABC 2.0 - Reference class

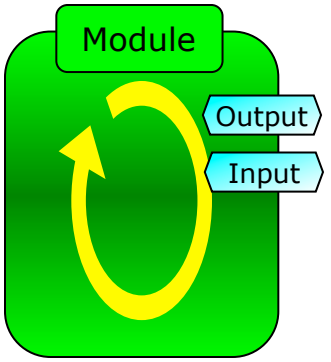
- Preserves pointer on any DABC2 object
- Prevents object to be deleted until references are existing
  - of major importance in multithread environment
- Released automatically when leaving method/loop context
  - simplifies a lot shutdown sequence and exception handling
- Used in many places of DABC2:
  - memory buffer, command, parameter, ...

DABC 2.0 – input/output ports

- DABC v1
  - all ports provides input and output functionality
  - natural in case of socket connection
  - not true in case of files, hardware devices

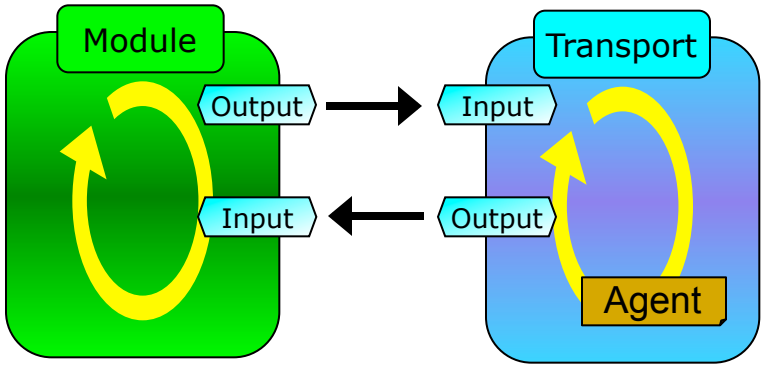


- DABC v2
  - input and output port classes were separated
  - one could bind ports to reuse same transport



DABC 2.0 – new transport concept

- DABC v1
  - transport runs in separate thread
  - very optimized interface to provide max performance
  - mostly no mutexes were involved
  - very complex in initialization and destroyment
  - a lot of duplications in transport classes



- DABC v2
  - reuse module functionality
  - transport for output port is just special module with input port
  - immediately clarifies and simplifies implementation

DABC 2.0 - better multi-node handling

- No more strict master-slave paradigm
- Possibility for direct communications between dabc2 nodes
- New connection manager
  - flexible connection scheme
  - possibility of automatic reconnect

DABC 2.0 - Record class

- Container for arbitrary list of attributes and values
- Provides direct mapping to/from xml
- Use by three classes:
  - Command
  - Parameter
  - Config

DABC 2.0 - many other features

- Decoupling control interface from DABC internals
  - let us provide different control back ends
  - several controls can run simultaneously
- More simple and solid memory management
- DABC binary file format
- URL syntax for data sources / data drains
- More flexible xml syntax

XML file example

```
<?xml version="1.0"?>
<dabc version="2">
  <Context name="Worker">
    <Run>
      <lib value="libDabcMbs.so"/>
      <func value="StartMbsCombiner"/>
      <logfile value="combiner.log"/>
    </Run>
    <MemoryPool name="Pool">
       <BufferSize value="65536"/>
       <NumBuffers value="100"/>
    </MemoryPool>
    <Module name="Combiner">
       <NumInputs value="3"/>
       <NumOutputs value="1"/>
       <InputPort name="Input0" url="mbs://lxi009/Transport" queue="5"/>
       <InputPort name="Input1" url="mbs://lxi010/Transport" queue="5"/>
       <InputPort name="Input2" url="mbs://lxi011/Transport" queue="5"/>
       <OutputPort name="Output0" url="mbs://Stream" queue="5"/>
       <OutputPort name="Output1" url="lmd://combiner.lmd?size=128" queue="5"/>
     </Module>
  </Context>
</dabc>
```

DABC 2.0 release

- Since 15.04.2013 official development brunch
- Provides plugins for:
  - MBS
  - HADAQ
  - EPICS
  - InfiniBand
  - DIM
  - RFIO

DABC 2.0 - outlook

- Very flexible
  - can be applied for most DAQ tasks
  - any analysis code can be integrated as well
- Actual application – TRB3
- By request BNet will be reinvented again
  - ~2 months of work