

NSCLDAQ: The Past, Present, and Future

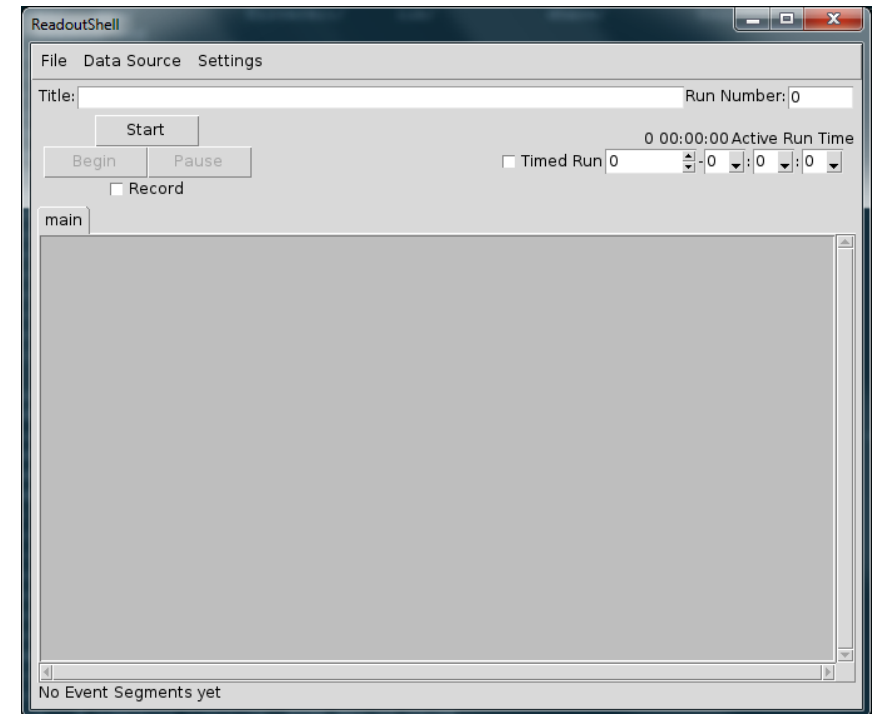
Jeromy Tompkins

Outline

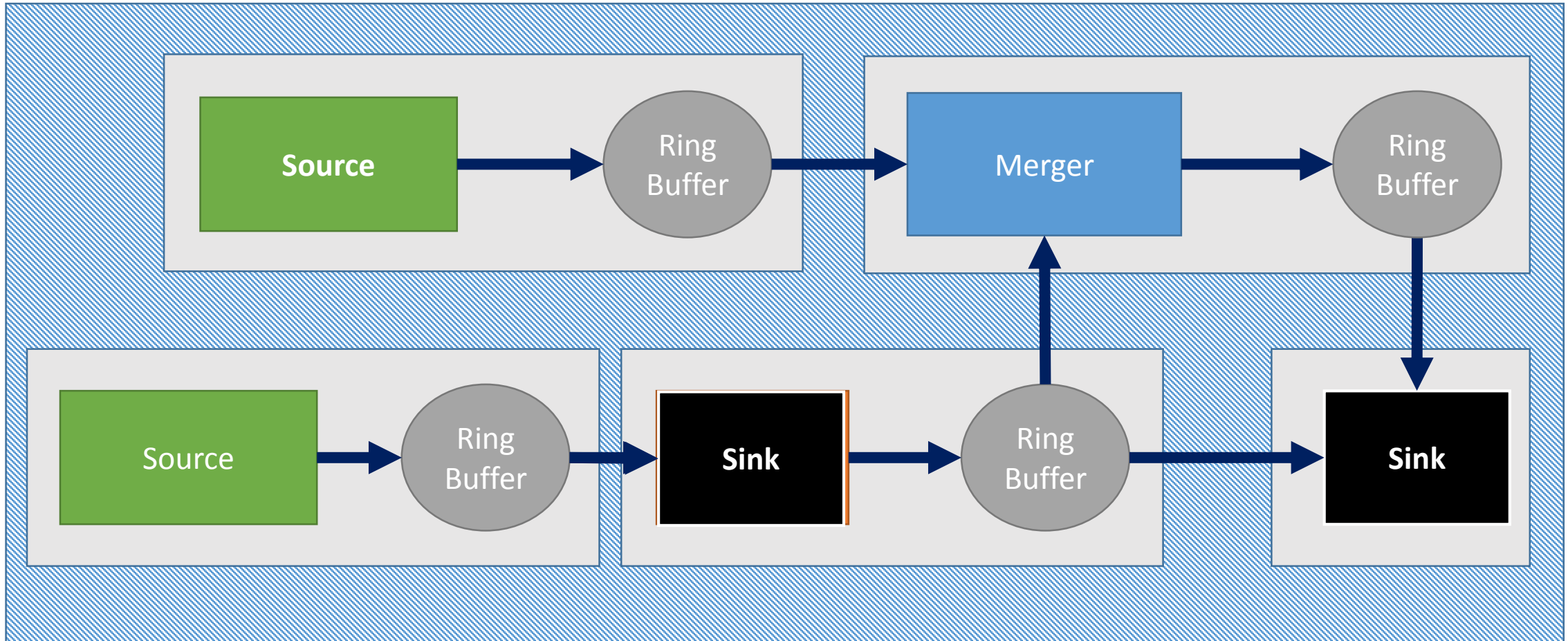
- What is NSCLDAQ?
- The need for NSCLDAQ 11.0
 - Why do we need a new major version?
 - Major features in 11.0
 - A case study – Unification of CAESARDaq and S800Daq
 - So what?
- Procedural tid bits
- Towards the future

What is NSCLDAQ?

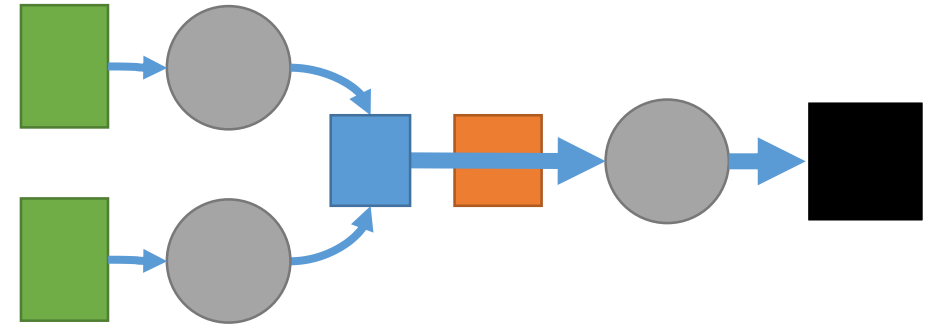
- Suite of applications and services
- Provides data flow management and run control
- Flexible
- Extensible
- Composable
- A set of drivers



NSCLDAQ Concepts



The suite...



Run Control

ReadoutShell

Data Sources

VMUSBReadout
CCUSBReadout
Production Readout
s800toring

Filters

BufferToRing
compatibilitybuffer
convert10to11
Filter framework

Merging

EVB tcl package
glom
teering
unglom
ringFragmentSource
offlineorderer
startOrderer

Data Sinks

dumper
ScalerDisplay
sclclient
eventlog

Buffering/Streaming

ringbuffer
(RingBuffer API)

stdintoring
ringtostdout

We are about to move from version 10.2 to 11.0 !!

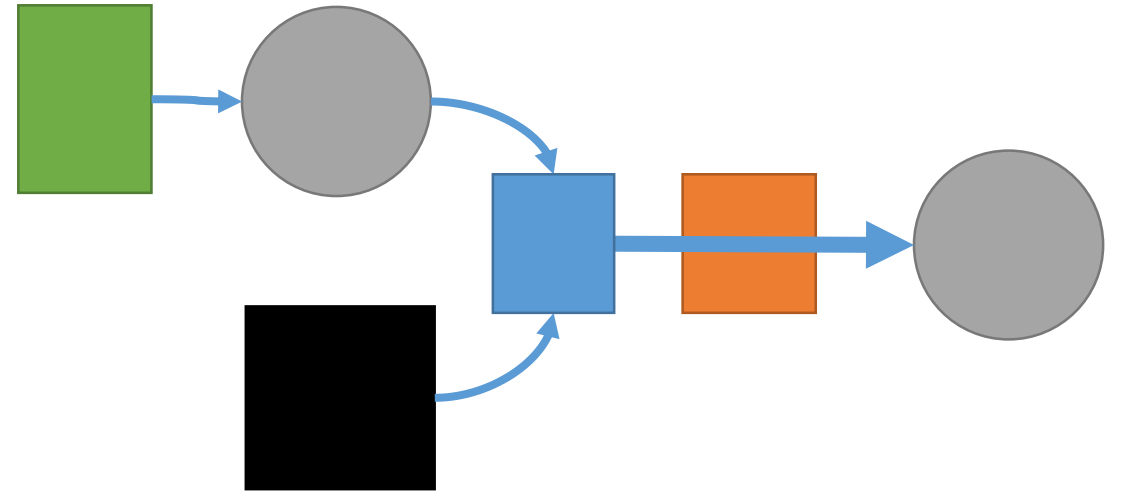


Why NSCLDAQ 11.0?

- Increased demand for event building
- Need to simplify the unification of separate DAQs
 - Data flow management
 - Run control

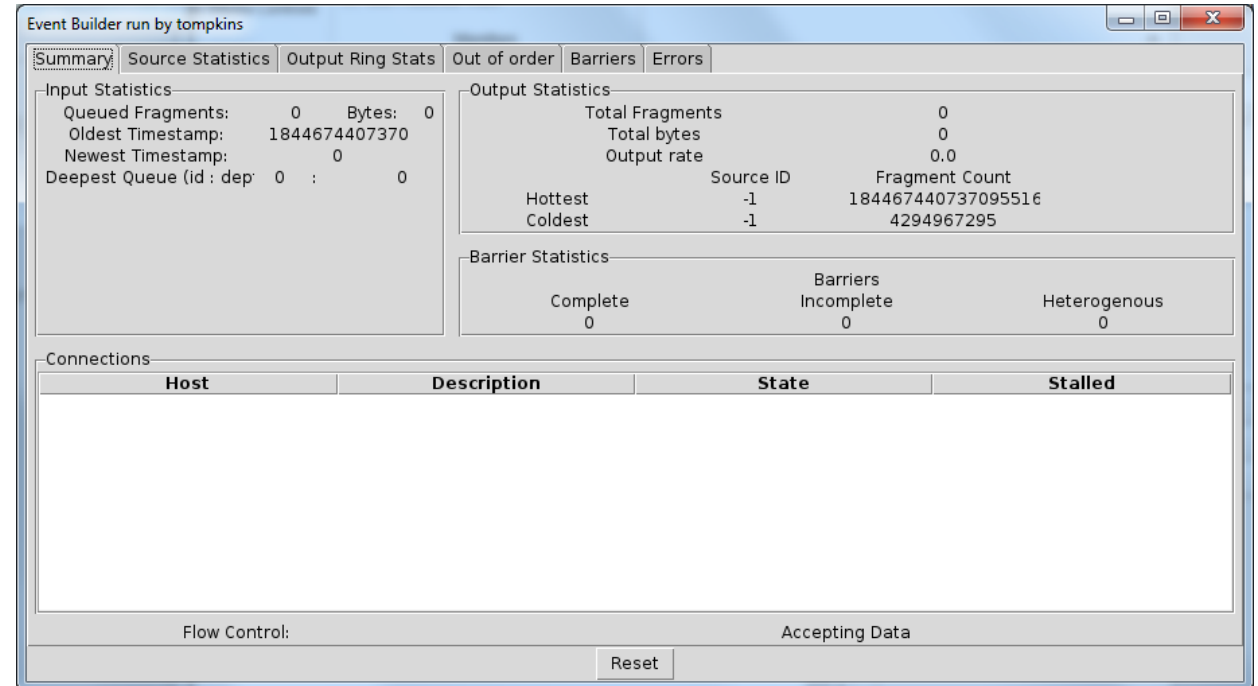
Major Features

- **Improved support for event building**
- Updated data format
- Revamped ReadoutGUI
- Filter framework



Event Builder Improvements

- Diagnostics
 - Input and output rates
- Error detection
 - Duplicate timestamp values
 - Out-of-order timestamps
 - Late fragments
 - Incomplete barriers
- Input flow control
- Simplified setup
- Offline orderer



Data Format – Addition of the “body header”

- Ring items continue to be standard NSCLDAQ data format
- Adds ability to embed timestamp and source id into data.

10.2

Header	Size (bytes)
	Type
Body	Data ...

11.0

(no body header)

Header	Size (bytes)
	Type
Body Header	Size (bytes) = 0
Body	Data ...

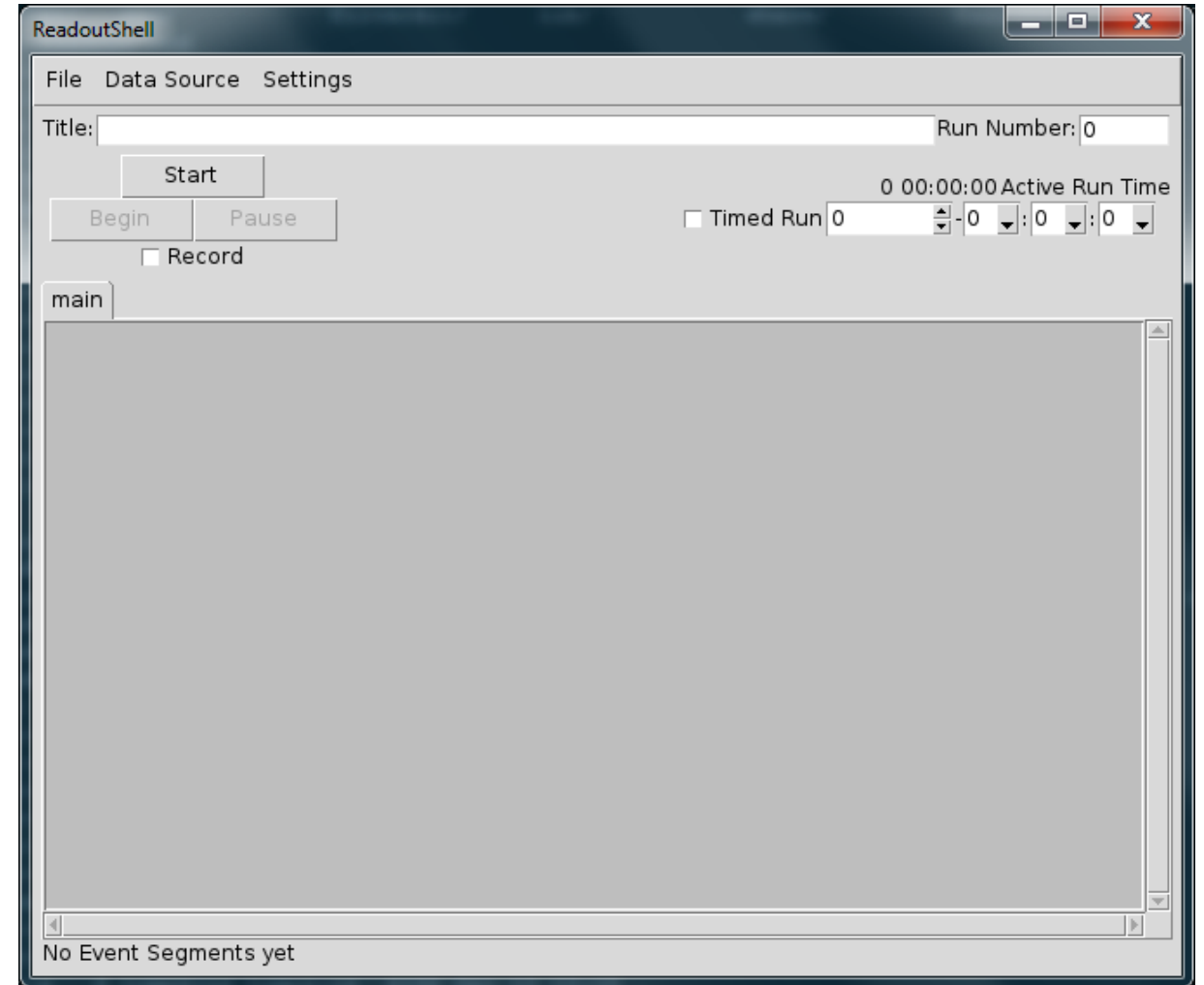
11.0

(with body header)

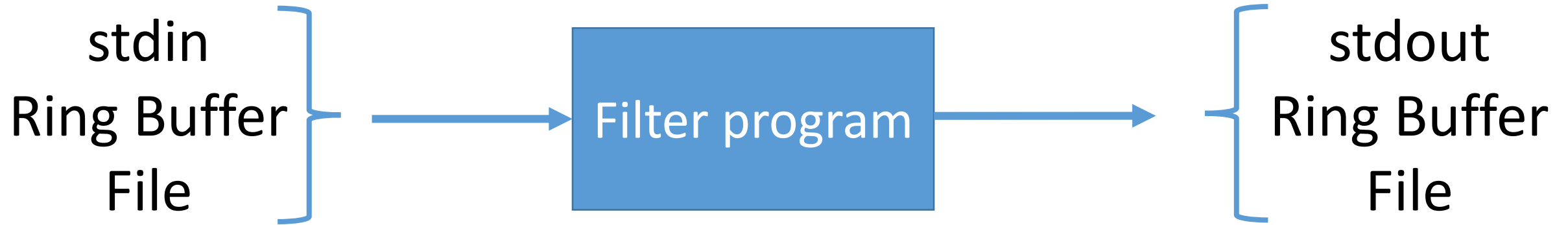
Header	Size (bytes)
	Type
Body Header	Size (bytes) = 20
	Timestamp
	Source id
	Barrier type
Body	Data ...

The 11.0 ReadoutGUI

- Simplified system building
- Controls an arbitrary subcomponents
- Extensible
- Remote controllable

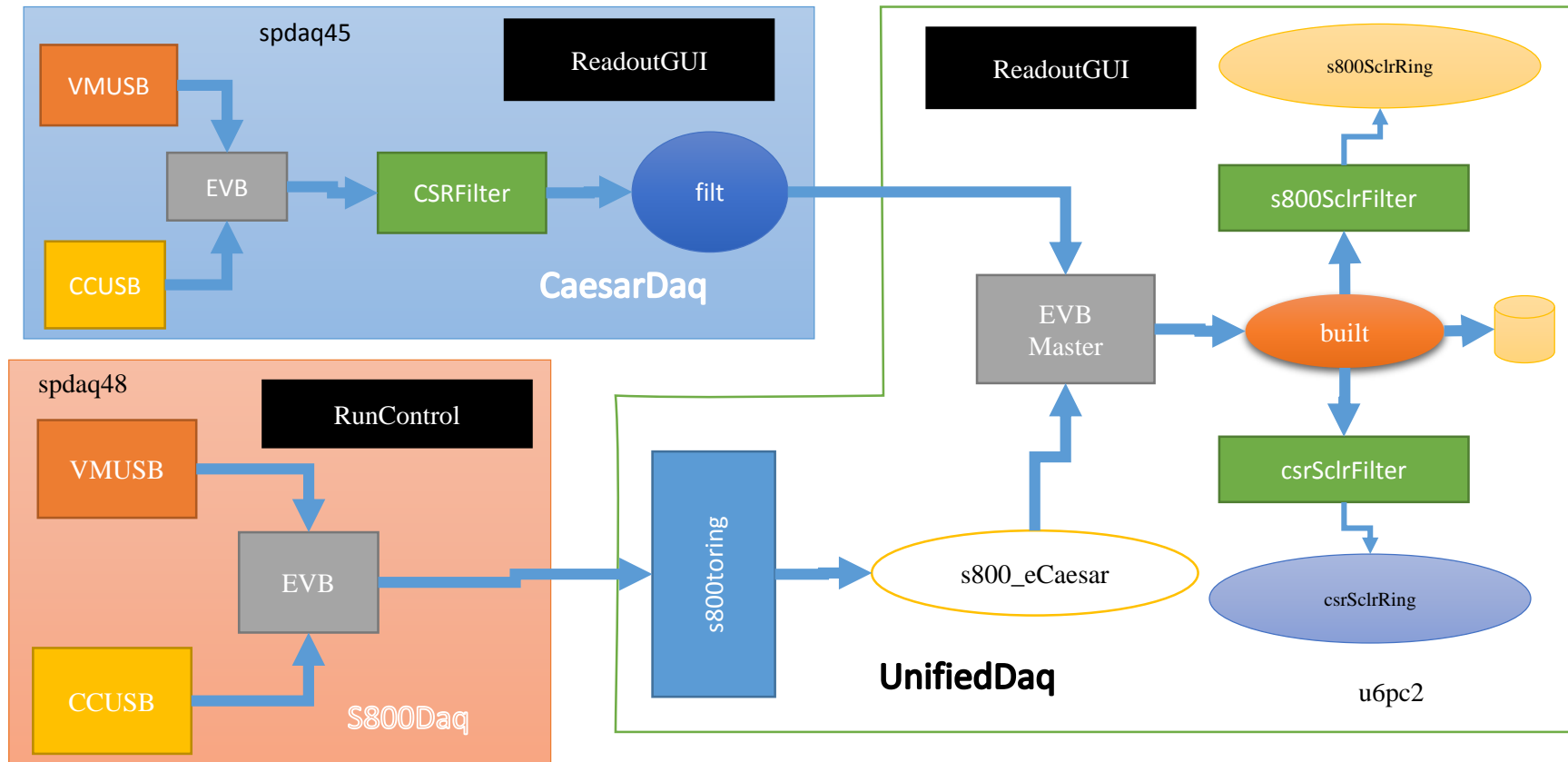


Filter Framework



- Streamlines development of online and offline tools
- Skeleton provides boilerplate code
- ROOT compatible!
- USES: data formatting, integrity checker, analysis platform, “traffic control”

The Final Product



So What?

- “Current” link will change to point to 11.0.
 - Make sure your system still works after this changes
- Earlier releases will not see new development
 - May be asked to upgrade to 11.0 to fix bugs if running an older system

Don't Panic!

- All previous releases will remain accessible.
- Compatibility software will exist for converting 8.x and 10.x data to 11.0 formats
- Documentation is available at docs.nscl.msu.edu/daq
- We are here if you still have questions.

How do I...

- Request a new feature or new device support?
 - Contact the DAQ committee. [translation: email Sean Liddick (chair)]
- Contribute a new feature to NSCLDAQ?
 - Ultimately needs to go through the DAQ committee, but feel free to chat with Ron or myself first.
- Report a bug?
 - High urgency (e.g. beam on target) → Phone call
 - Normal urgency →
 - File a bug against NSCLDAQ at <https://swdev-redmine.nscl.msu.edu>
 - Email tompkins@nscl.msu.edu or fox@nscl.msu.edu
- Find documentation?
 - Go to docs.nscl.msu.edu/daq
 - `man -M /usr/opt/daq/11.0/share/man`

Towards Versions 11.1, 11.2, 11.3, ... , 12.0

What do you need that you don't have now in NSCLDAQ 11.0?

Planned improvements

- Improve diagnostics
- Stability enhancements
- Improve support for legacy data formats
- Further simplification of experiment setup

Summary

- NSCLDAQ is a suite of software tools to manage data flow and run control.
- NSCLDAQ 11.0 is upon us and brings a host of improvements for event building
- Flexibility is built in to meet the wide range of experimental needs at the NSCL and beyond
- Future of NSCLDAQ is in your hands...we're listening.

Thank you.

- Ron Fox
- Sean Liddick