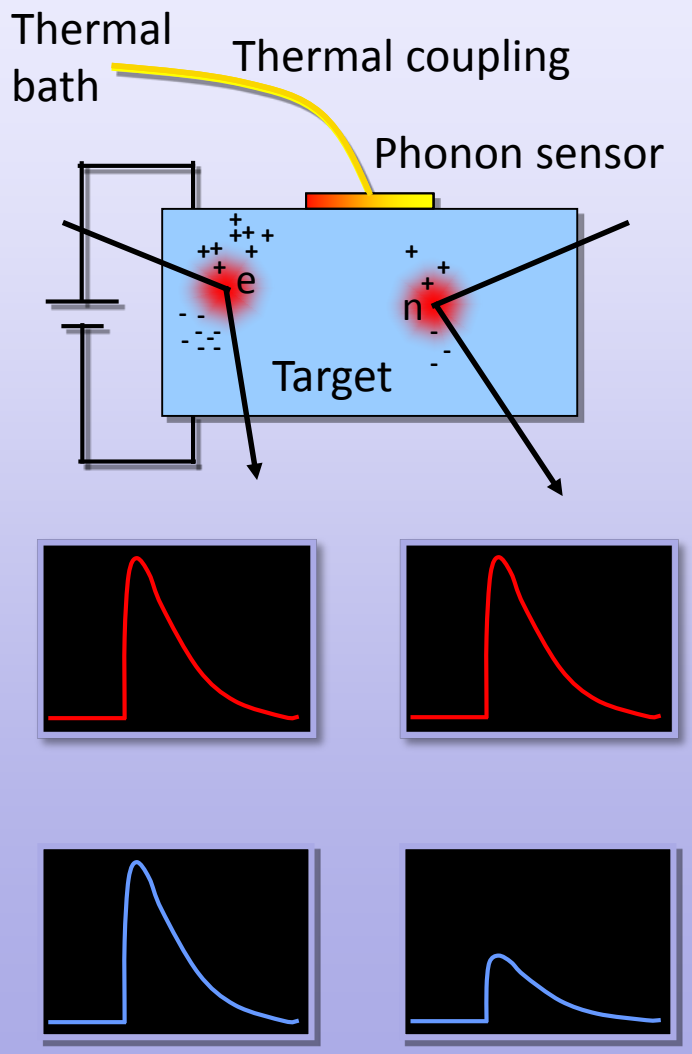
MIDAS for the SuperCDMS dark matter experiment

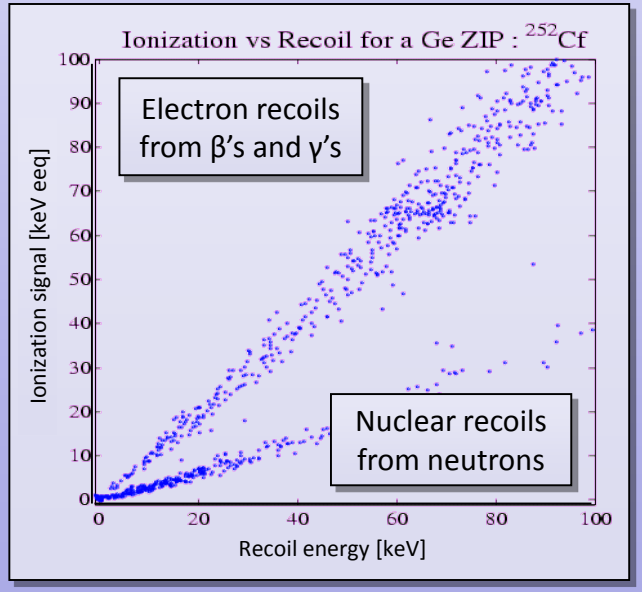
Scott Oser
MIDAS workshop
July 15, 2015

SuperCDMS Technology

Operation Principle

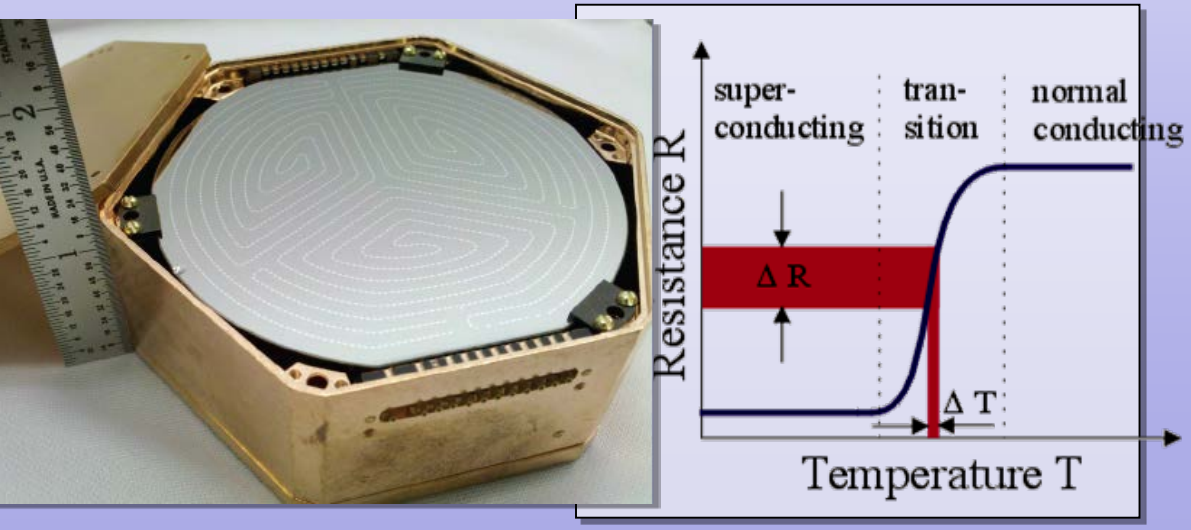


- High-purity Ge and Si crystals with readout of athermal phonons (using superconducting transition edge sensors) and ionization signals
- Electron recoil (ER) events produce more electron-hole pairs in semiconductor than nuclear recoils (NR) events do. Measure charge signal to discriminate between signal (NR) and background (ER)



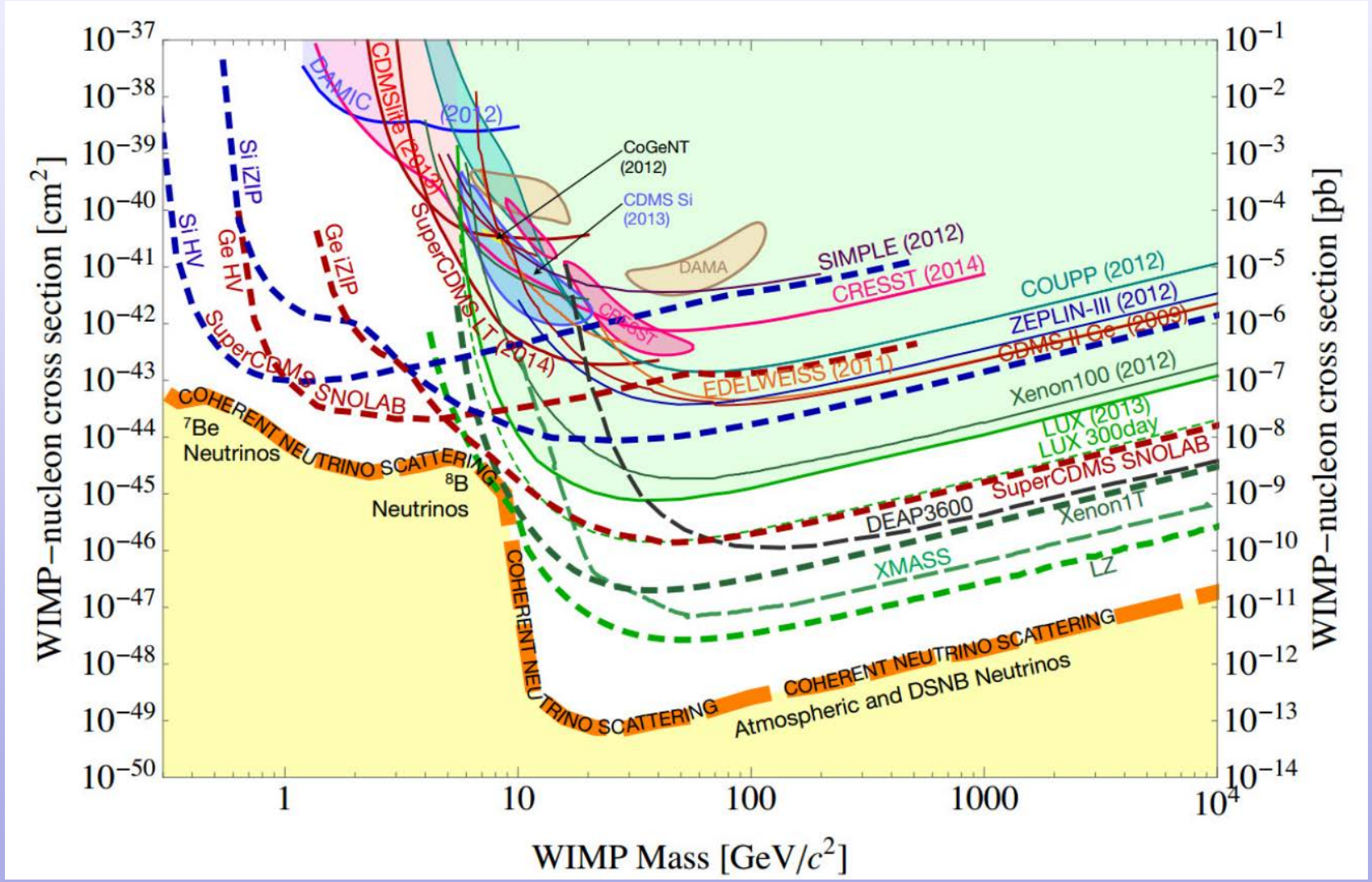
SuperCDMS upgrade: SNOLAB

- Previous generation:15 detectors, ~10kg mass in operation at Soudan in northern MN (2012-2015)
- Next generation experiment approved for SNOLAB: ~50kg mass using larger crystals and improved detector design. Large cryostat allows for future upgrades.
- Optimized for low energy threshold, low mass WIMPs



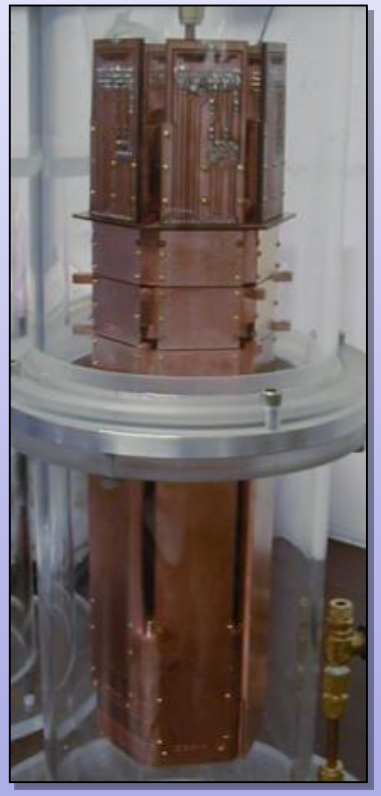
Background rejection based on ionization/phonon signal, plus timing cuts on phonon signals

Projected sensitivity



Warm Electronics Specification

Functional unit: one tower, containing 6 crystals. Full experiment will contain ~8 towers, or ~50 crystals.



Data type: digitized waveforms for 12 phonon channels (625 kS/s) and 4 ionization channels (2.5 MS/s), per crystal.

16-bit digitizers

Expected data size per triggered detector: ~140KB per detector read out

Trigger rate: ~5 Hz per detector for calibrations, near zero for WIMP search

Detector Control and Readout Card

One card per crystal = ~50 total

Continuous waveform digitization with 3.35 s circular buffer

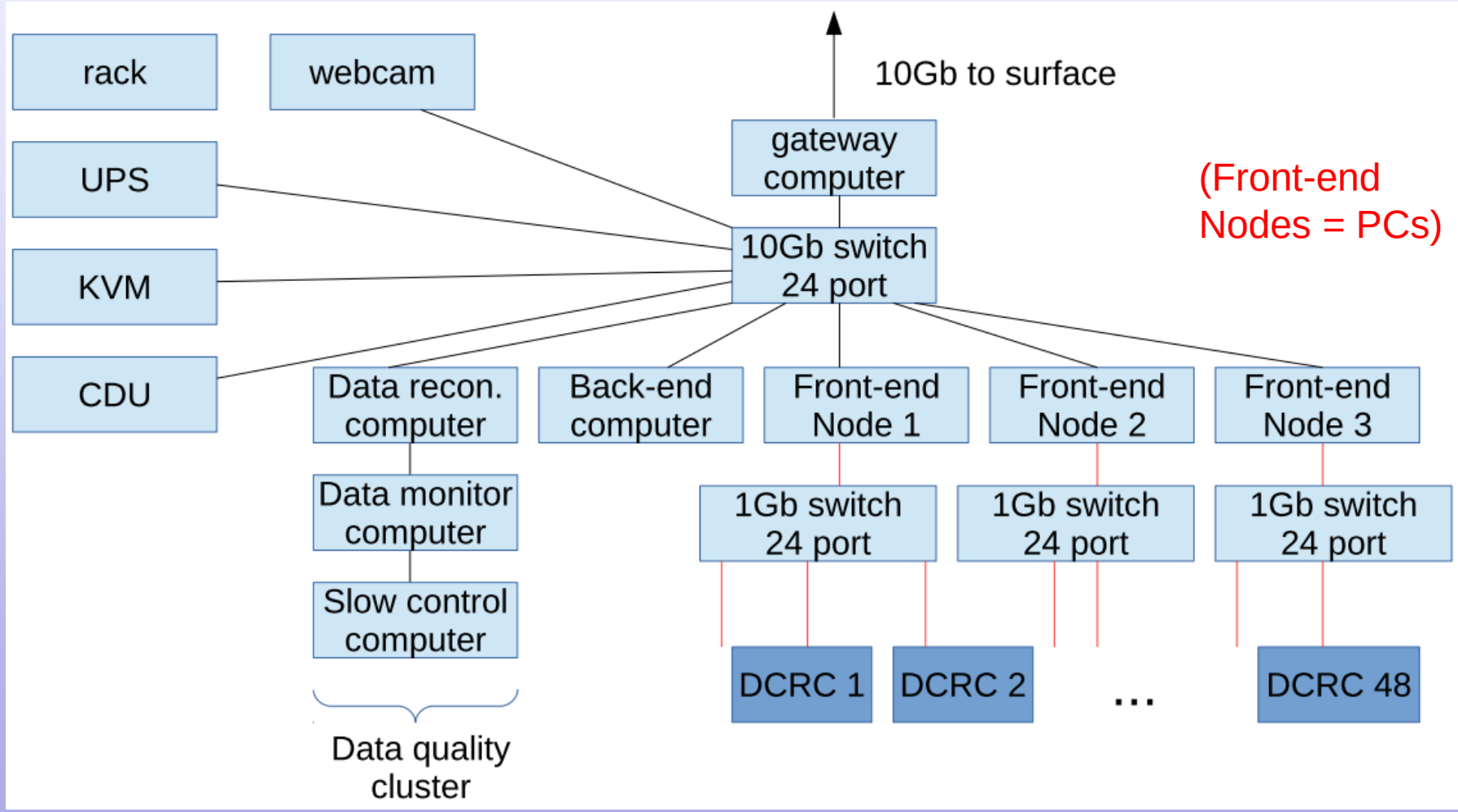
Internal card-level triggering on all channels (Level 1 trigger). To be read over ethernet.

Separate trigger buffer records locations in data circular buffer where triggers occurred. DAQ must first retrieve the list of card-level triggers over ethernet, then choose which to act on (Level 2 trigger---a software module), and finally read the waveforms.

TCP/IP communications, power over ethernet

Board also contains on-board test signal generator, LED pulser driver (for discharging crystals), and readout of voltages/temperatures on boards.

Back-end hardware



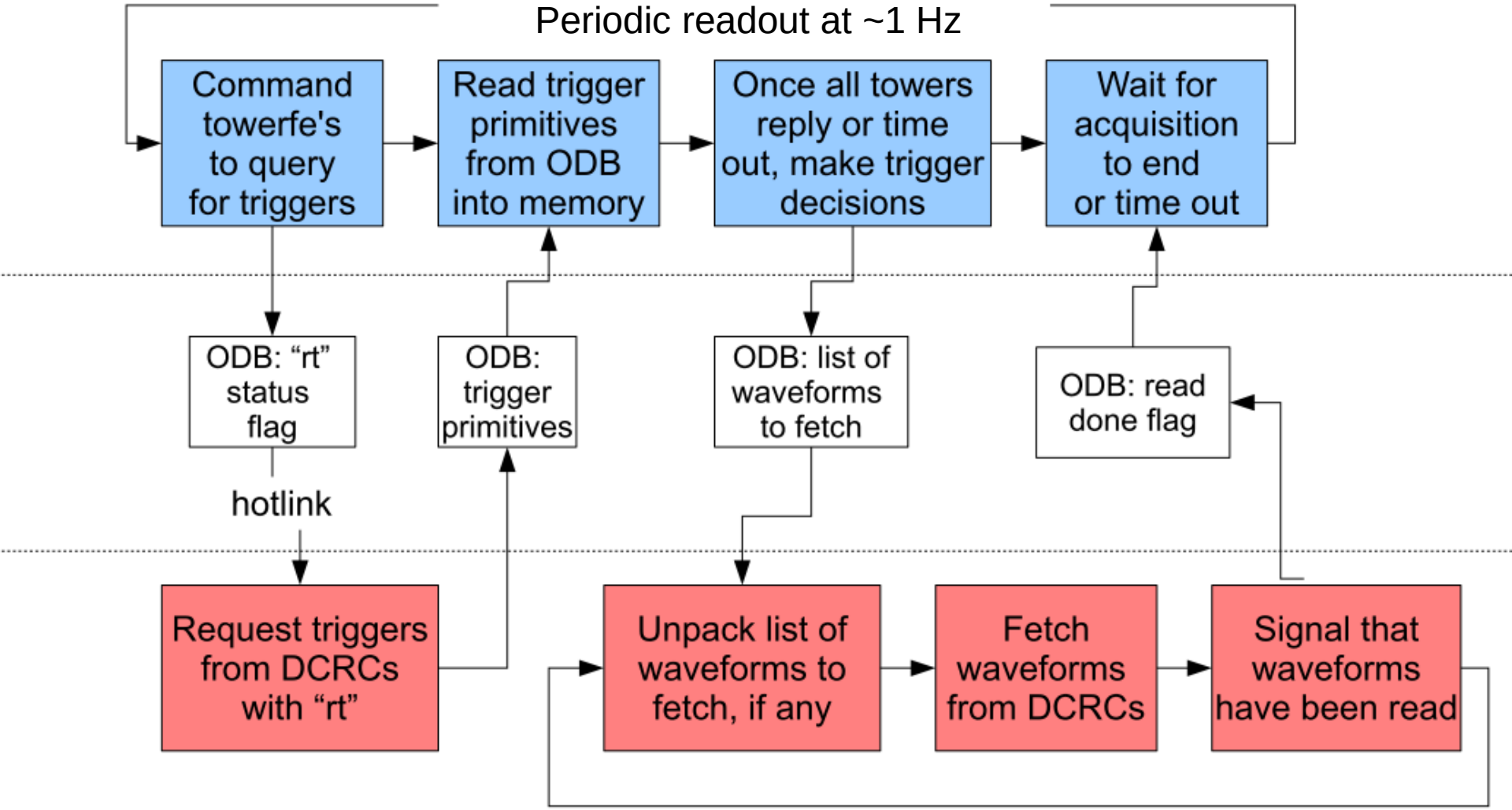
Triggering

No fast hardware trigger exists. Basic trigger is implemented in software, making use of the very long circular buffer memory to allow slow trigger decisions to be made. We plan to pass trigger information between front-ends using the ODB.

Basic main triggering modes:

- **"Normal" readout mode**: poll all detectors at ~1 Hz seeing if any have generated local on-board triggers. Read out locations in circular buffer where triggers occurred, make trigger decisions in DAQ, and then request waveforms from all detectors at that time index. Low rate trigger.
- **"Selective" readout mode**: read out any waveforms on any detector that triggered, but don't read out non-triggering detectors. Mostly used for calibration running at higher rates.
- **"Intra-tower coincidence trigger"**: if two detectors inside one tower both recorded a trigger within a defined time window, read out both waveforms.

Central trigger processor: a separate process that only talks to ODB



Tower front-end: talks to hardware directly, probably runs on different node than trigger processor

Periodic readout at ~20ms intervals: most of the time there will be no waveforms to fetch, and so nothing to do.

Passing trigger data through the ODB

Tower Front-end

Runs on front-end PC: periodic readout with 20ms period

Main loop: query ODB to see if waveforms have been requested.

- If so, fetch waveforms from DCRCs and stuff into data bank.

Trigger handler: when activated by hotlink from Trigger Module:

- Read raw trigger info from DCRCs
- Write trigger info into ODB as a long character string (~7kB)
- Set a "finished" flag in the ODB

Trigger Module

Runs on back-end PC: periodic readout with 1s period

Main loop (once a second)

- Instruct tower front-ends to get triggers by writing to a hotlinked key
- Wait for all front-ends to return trigger info
- Decide which triggers to request waveforms from
- Write list of waveforms to read into ODB
- Wait for all towers to finish

Why this design?

Central trigger processor doesn't have to talk to hardware, only to ODB.

Centralized sequencing of trigger requests and waveform reads: code will never attempt to read new triggers until older waveforms are acquired. All towers will supply new trigger information at the same time, making trigger decision algorithm much simpler.

Most important design driver: we knew how to do it.

Performance Notes

Tests of throughput running multiple front-ends on remote nodes, with each front-end nominally providing 5.8MB/s of data.

# of front-ends	Reported rate from MIDAS	Total data rate	Rate per tower front-end
5 (1 node)	1.0 Hz	28 MB/s	5.6 MB/s
5+5 (2 nodes)	1.0 Hz	62.5 MB/s	6.25 MB/s
5+5+5 (3 nodes)	1.0 Hz	83.9 MB/s	5.6 MB/s
5+5+7 (3 nodes)	0.8 Hz	75 MB/s	4.4 MB/s
5+5+5+5 (4 nodes)	0 Hz	0	0

Throughput hits a wall when total data rate approaches 75% of measured network capacity. The wall isn't 100% understood, but seems to be due to the network being jammed and delaying the subsequent request from the Trigger Module to initiate the next round of triggers.

Still, the throughput seems adequate for now.

Plea for feedback

Is there a better way to pass information between front ends and the trigger module?

For example, would using buffers be a better way to exchange trigger information than going through the ODB?

ODB Lock Issues

We want to generally lock detector settings while running, but certain settings may need to be automatically adjusted or reset during a run.

What we really want is a way of allowing certain processes to change settings or scripts to change settings while blocking others. MIDAS sequencer doesn't seem to permit locking/unlocking, nor does mhttp/javascript.

Our kludgy solution so far has been to create a "lock manager" that hot links an integer "lock setting", and automatically locks and unlocks parts on the ODB in response. For example, a sequencer script can write to this key and ask the lock manager to unlock certain pre-selected variables, then write to the key again to relock them when finished.

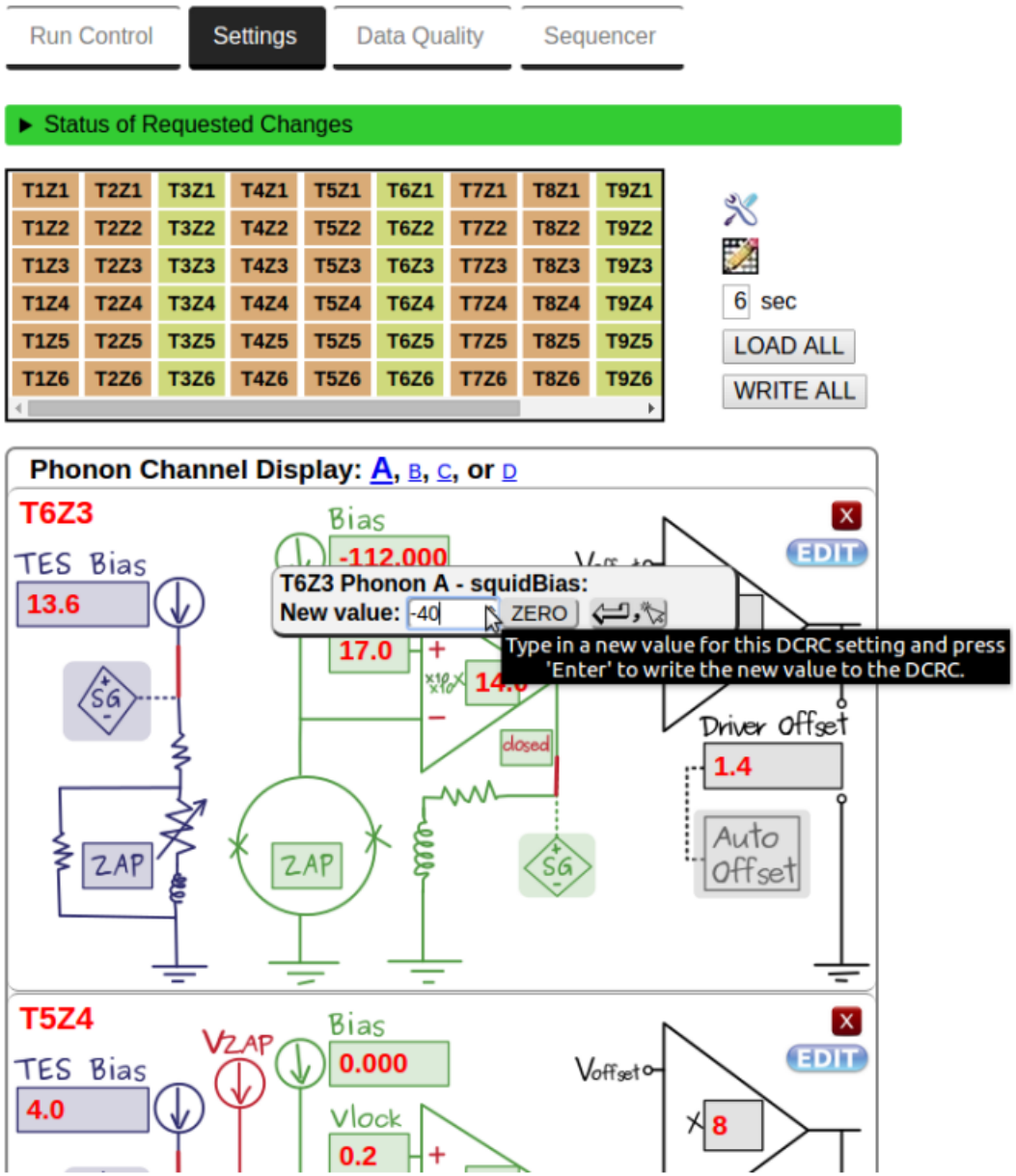
Any better solution?

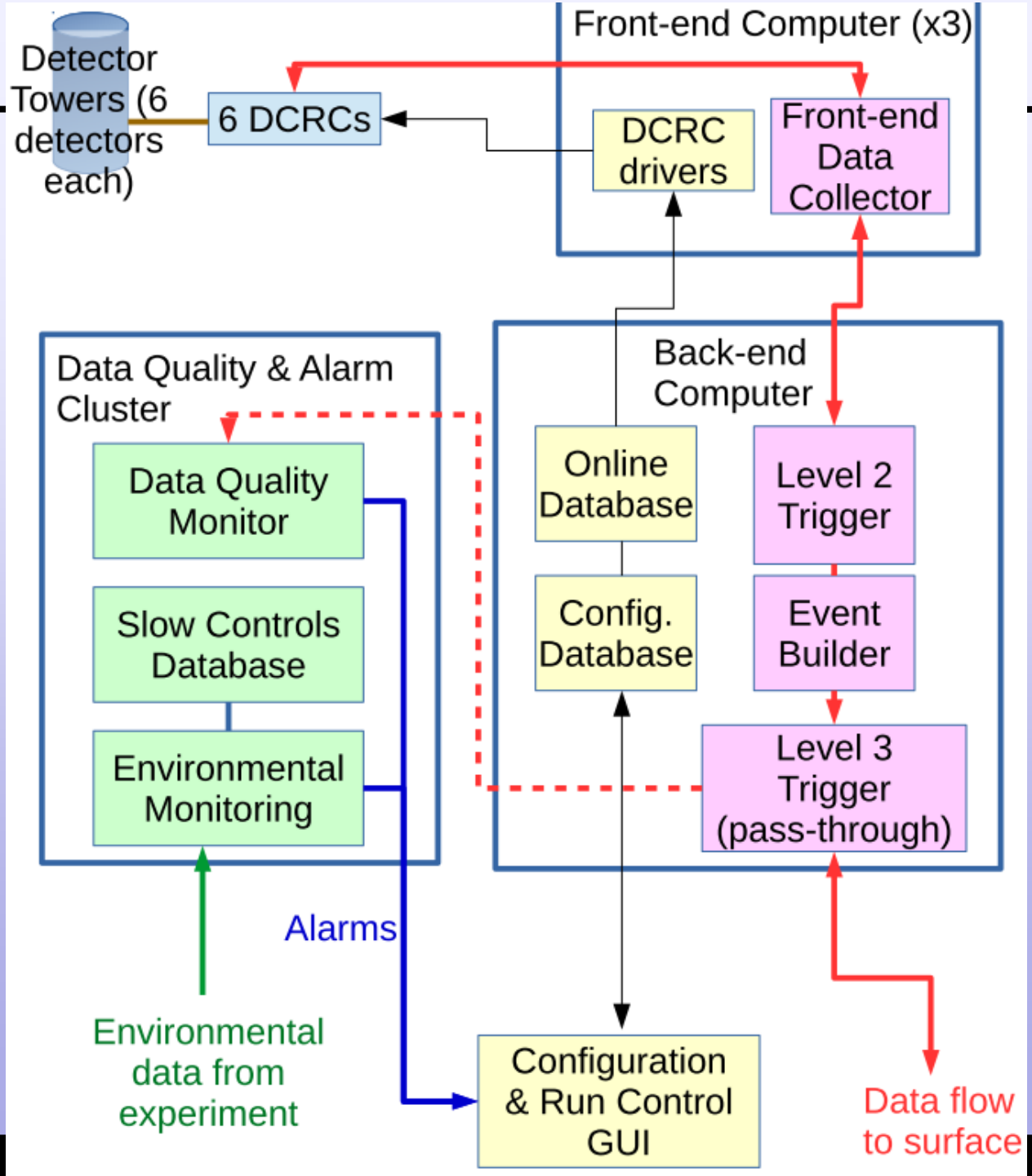
Configuration GUI

Custom javascript-based detector configuration editor.

We plan to connect this to a couchdb database similar to what DEAP uses.

Will likely be expanded to include a custom run control page.



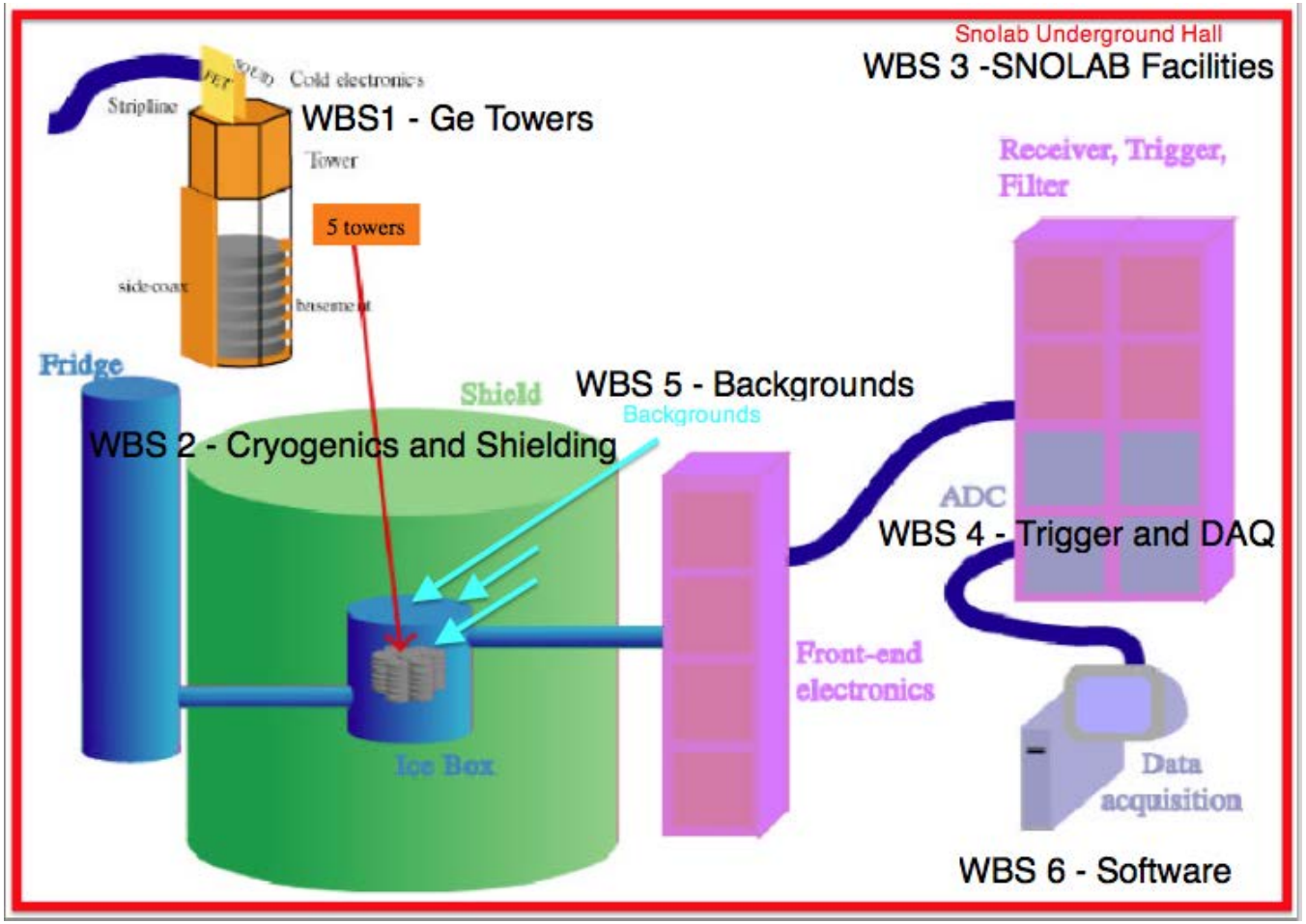

Front-end Computer (x3)
Detector Towers (6 detectors each)
6 DCRCs
DCRC drivers
Front-end Data Collector
Data Quality & Alarm Cluster
Data Quality Monitor
Slow Controls Database
Environmental Monitoring
Back-end Computer
Online Database
Level 2 Trigger
Config. Database
Event Builder
Level 3 Trigger (pass-through)
Alarms
Environmental data from experiment
Configuration & Run Control GUI
Data flow to surface

Detector Tuning GUIs

These detectors require a lot of interactive tuning, especially during development phase. Very far from the ideal of loading settings and starting run. Previously all done with custom LabView VIs. Our approach has been to write ROOT GUIs that link to MIDAS and can communicate with ODB. Sort of a hybrid of a MIDAS analyzer with run control and ODB hooks.

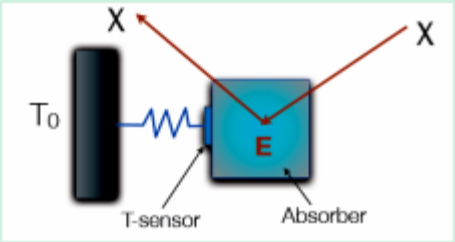
Backups

Schematic of SuperCDMS SNOLAB



Cryogenic Techniques at mK Temperatures

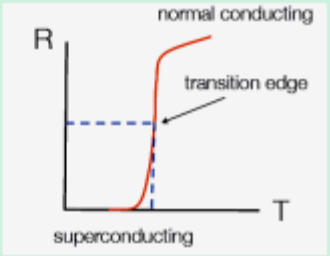
Need sensitivity (big ΔT) for small ΔE, so run at $T << T_c$



$$C(T) = \frac{\Delta E}{\Delta T} \propto T^3$$

Bolometric: ROSEBUD, EDELWEISS
Neutron transmutation-doped Ge (NTD-Ge) sensor +FET

Phonon Signal: CRESST, CDMS
Superconducting Transition Edge Sensors (TES)
+ SQUID readout



Backgrounds

Bulk electrons/gammas:

- **Reject with phonon/ionization ratio**
- **Gammas can scatter in multiple detectors**
- **Expose to ^{133}Ba source (356 keV gammas) to calibrate**

Surface electrons:

- **Imperfect charge collection lowers ionization yield**
- **Risetimes of phonon signal different for surface events**

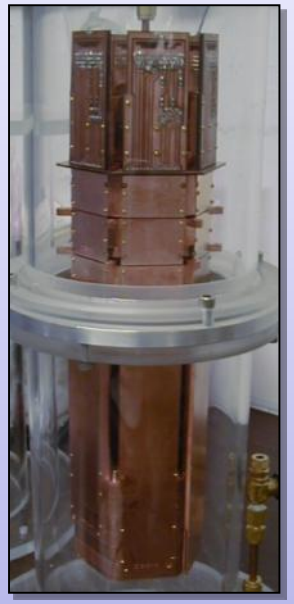
Neutrons:

- **Go deep and shield**
- **In future consider active neutron veto**

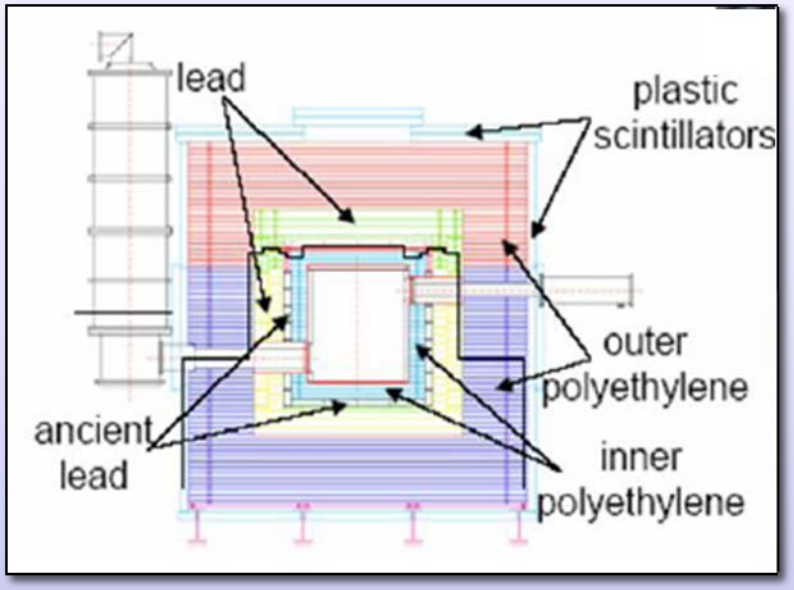
CDMS Technology

CDMS Setup

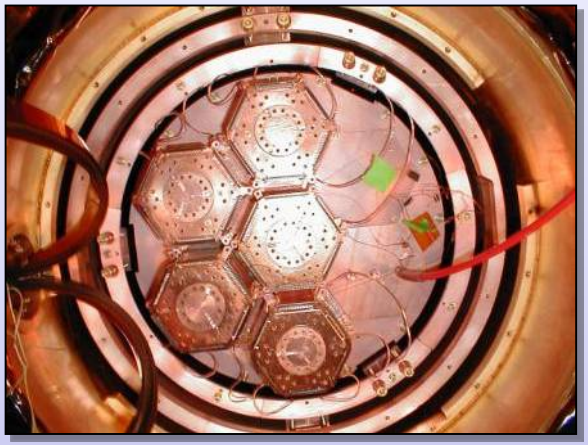
"Tower"

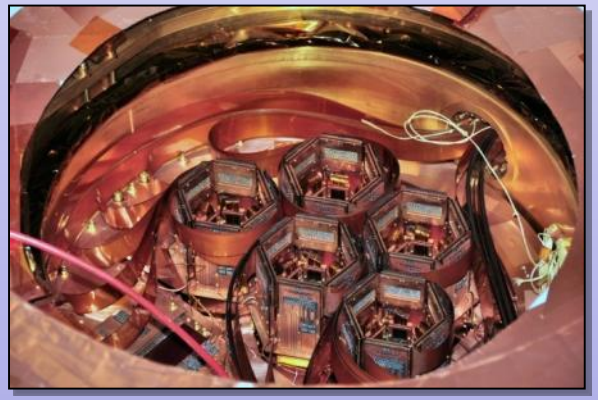


Stack of 6 detectors



5 Towers: ~5 kg Ge, 1 kg Si





Operated in Soudan Lab (Minnesota) 2006 – 2009

CDMS-II Analysis and Results

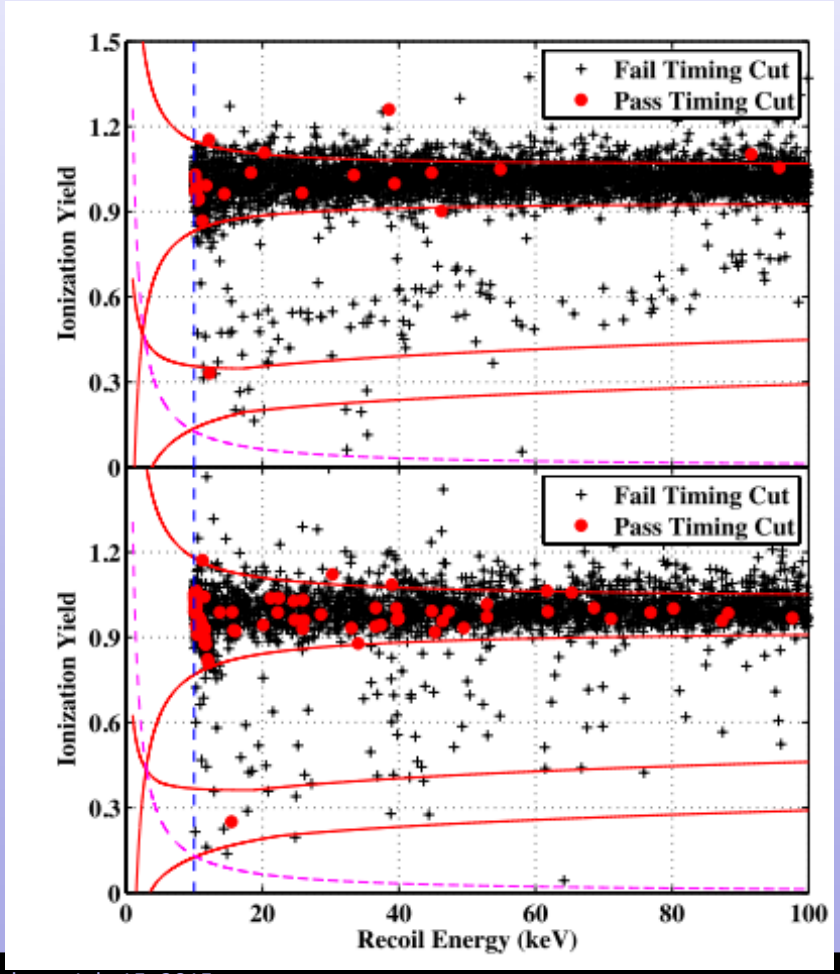
Published Data:

- CDMS-II data to September 2008
- Raw exposure: 612 kg days
- Analysis threshold: 10 keV
- Main analysis steps:
  - Cover signal region
  - Remove periods with bad detector performance
  - Determine position dependent calibration and timing performance
  - Remove multiple scatter & muon veto events
  - Remove surface events (timing)

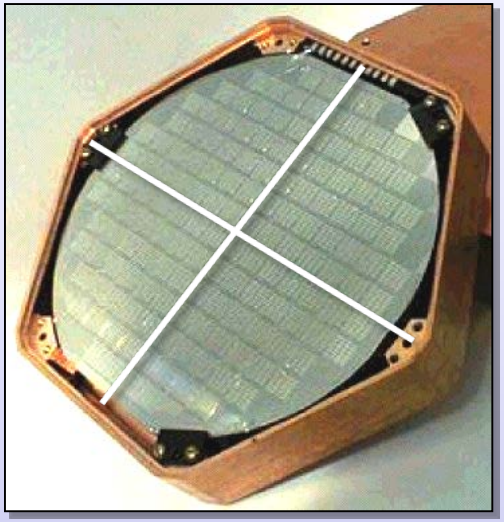
  0.9± 0.2 events expected
  - Open the box
- Two events observed!
  - P-value=23%
  - One looks like possibly misreconstructed surface event

Backgrounds:

- Cosmogenic neutrons: 0.04±0.04
- Bulk e/γ: 0.03-0.06
- Surface events: 0.8±0.1±0.2

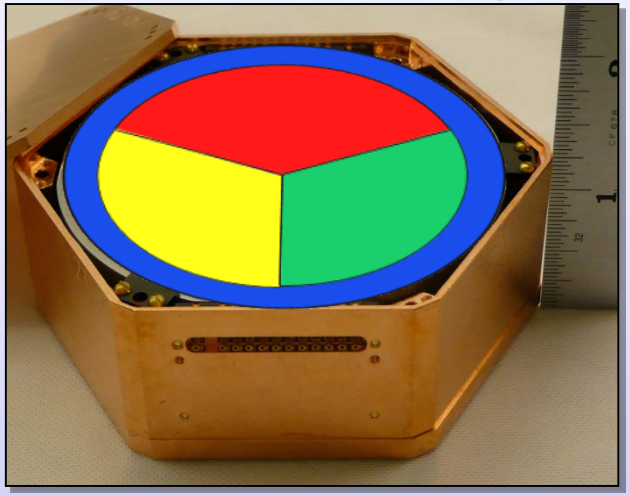


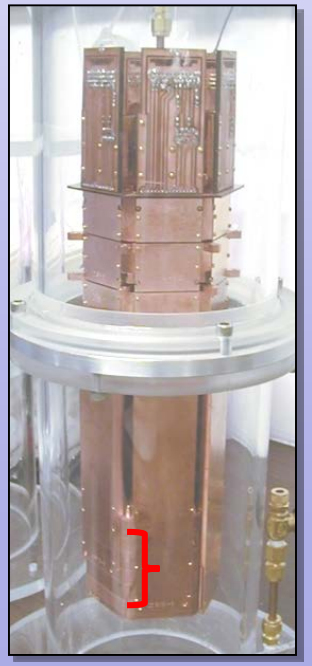
SuperCDMS



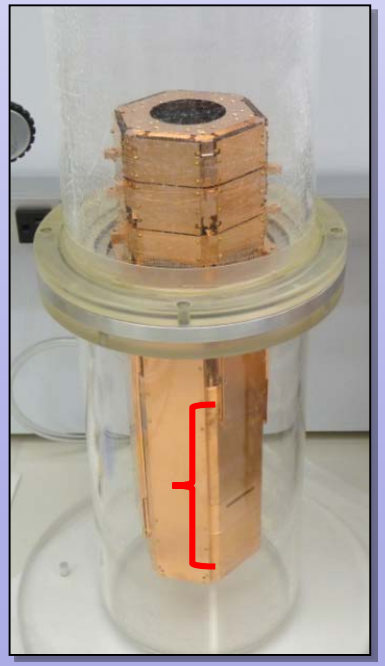
Larger detectors
(250 g → 630 g)

Improved sensor design





Five towers with 3 detectors of the iZIP design deployed at Soudan.



Medium term goal: Further increase mass/module; build 100-200 kg experiment at SNOLAB

Long term: ~ 1 ton