White Rabbit installation at Fermilab in the SBN program and ICARUS implementation

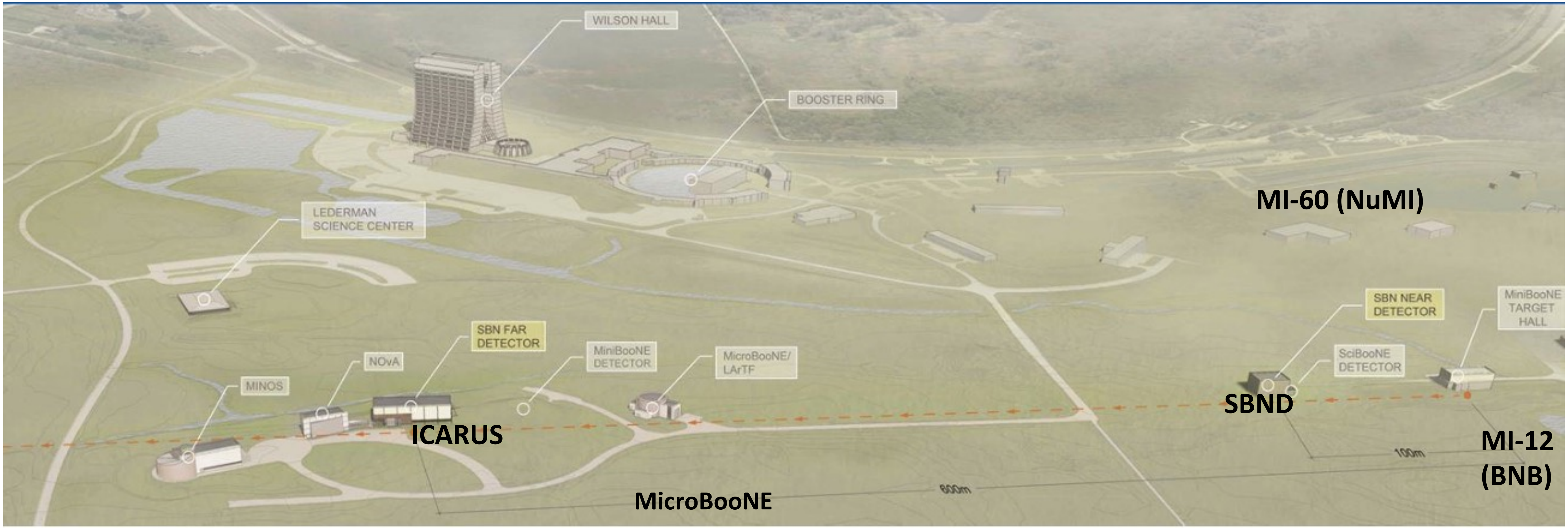
D. Torretta, Fermilab

Precision Timing Workshop

March 23, 2023

SBN campus

TWO beams: BNB and NuMI
TWO experiments: SBN and ICARUS



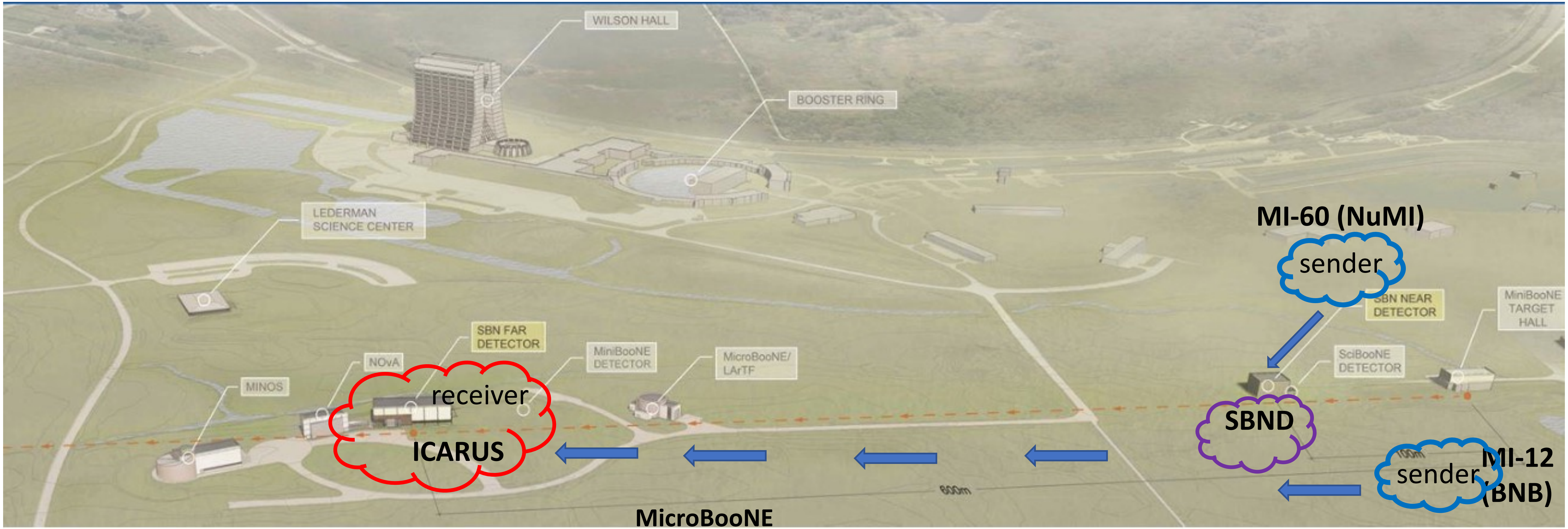
To distribute the beam signals to the experiments we implemented a fully deployed White Rabbit Network with sub-ns accuracy synchronization.

The “absolute” GPS timing in form of pps (pulse per second) signal will be provided to all SBN detectors and used as a reference for generating phase locked digitization clocks and stamping the beam gates and trigger signals.

Locations were already connected via an underground network of AD (Acceleration Division) SM/MM fibers.

SBN campus: location of WR switch/nodes

Installation done in 2017 by
W. Badgett, A.Fava & D.Torretta



Sender nodes (Linux) :
sbn-mi60 (PCIe)
sbn-mi12 (PCIe)

GrandMaster WR switch at ICARUS

Receiving nodes:
ICARUS: SPEXI/DIO in NI crate (PXIe card)
icarus-daq02 (SL7, SPEC/DIO PCIe card)
icarus-clk02 (Ubuntu, SPEC/DIO PCIe card)

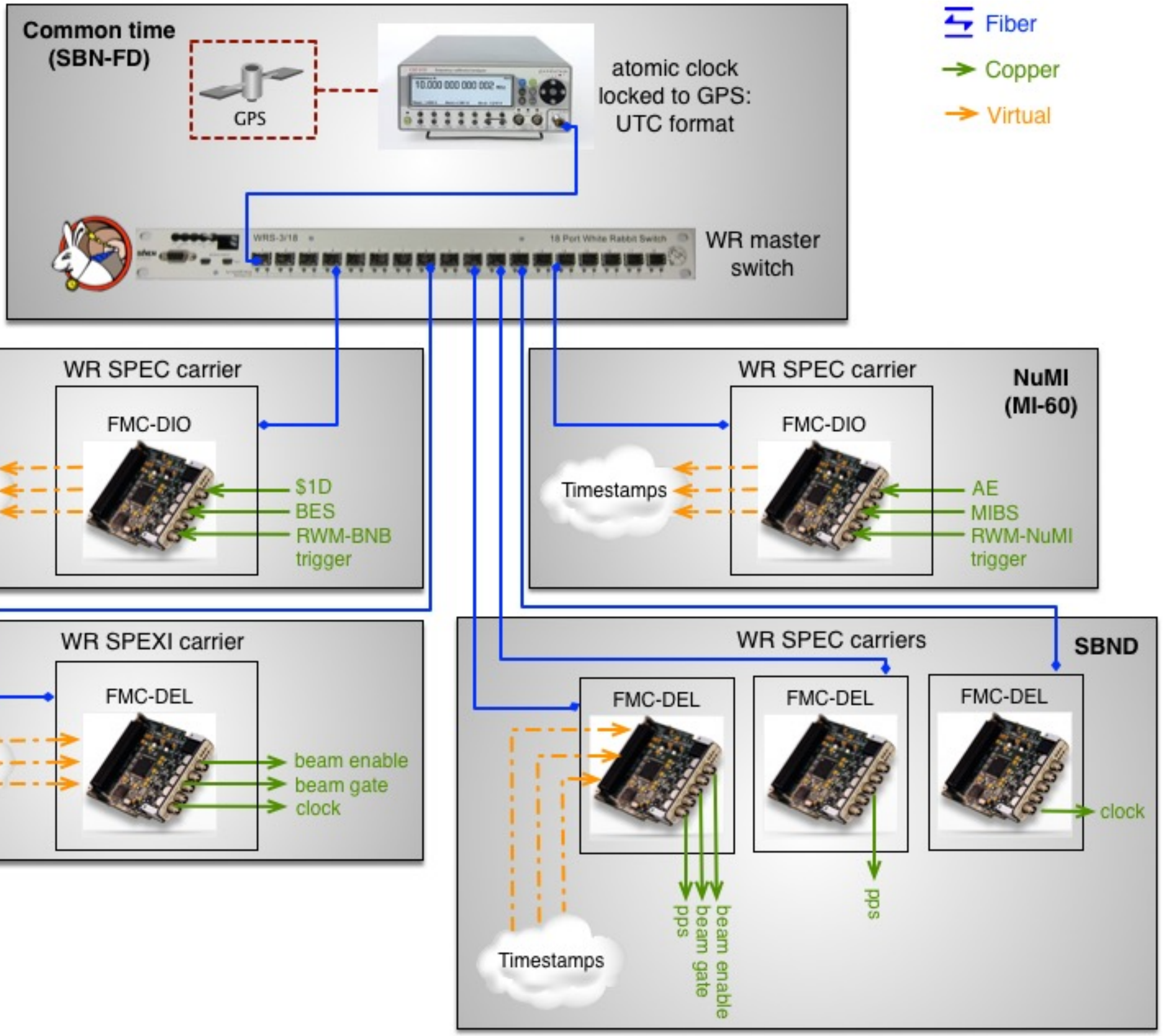
SBND : **tbd, not in the network at the moment**

Architecture of the White Rabbit network at Fermilab

WR GrandMaster switch
Installed in ICARUS Timing Rack
in West mezzanine

Server with SPEC/DIO

NI crate with SPEXI/DIO



Server with SPEC/DIO

SBND setup changed
over the years
(next week's talk)

Beam signals from MI-12 (BNB) and MI-60 (NuMI)

BNB

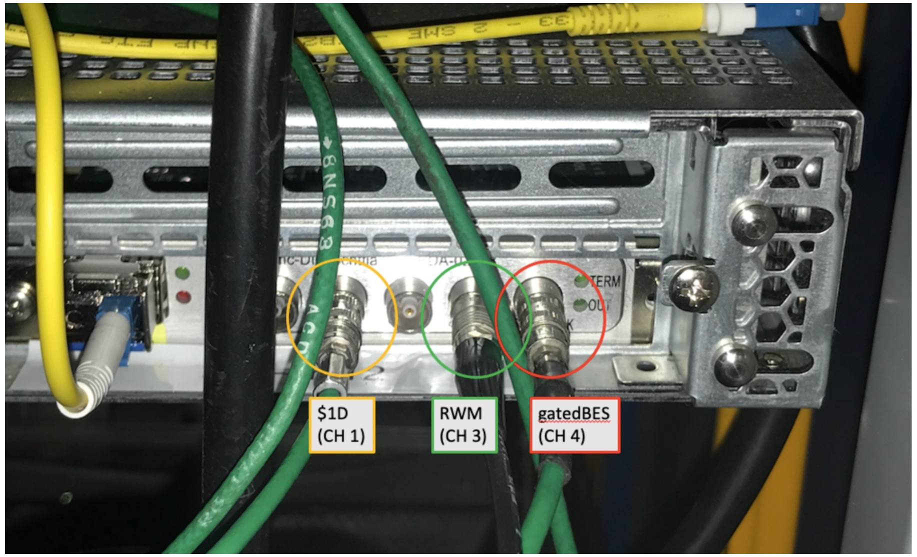
ID = 4 : gatedBES [EW 0.335 ms before proton on target]

ID = 1 : $1D [EW 35.7 ms before proton on target]

NUMI

ID = 4 : MIBS$74 [EW 1.788 ms before proton on target]

ID = 1 : $AD [EW 572 ms before proton on target]



WR node at MI12

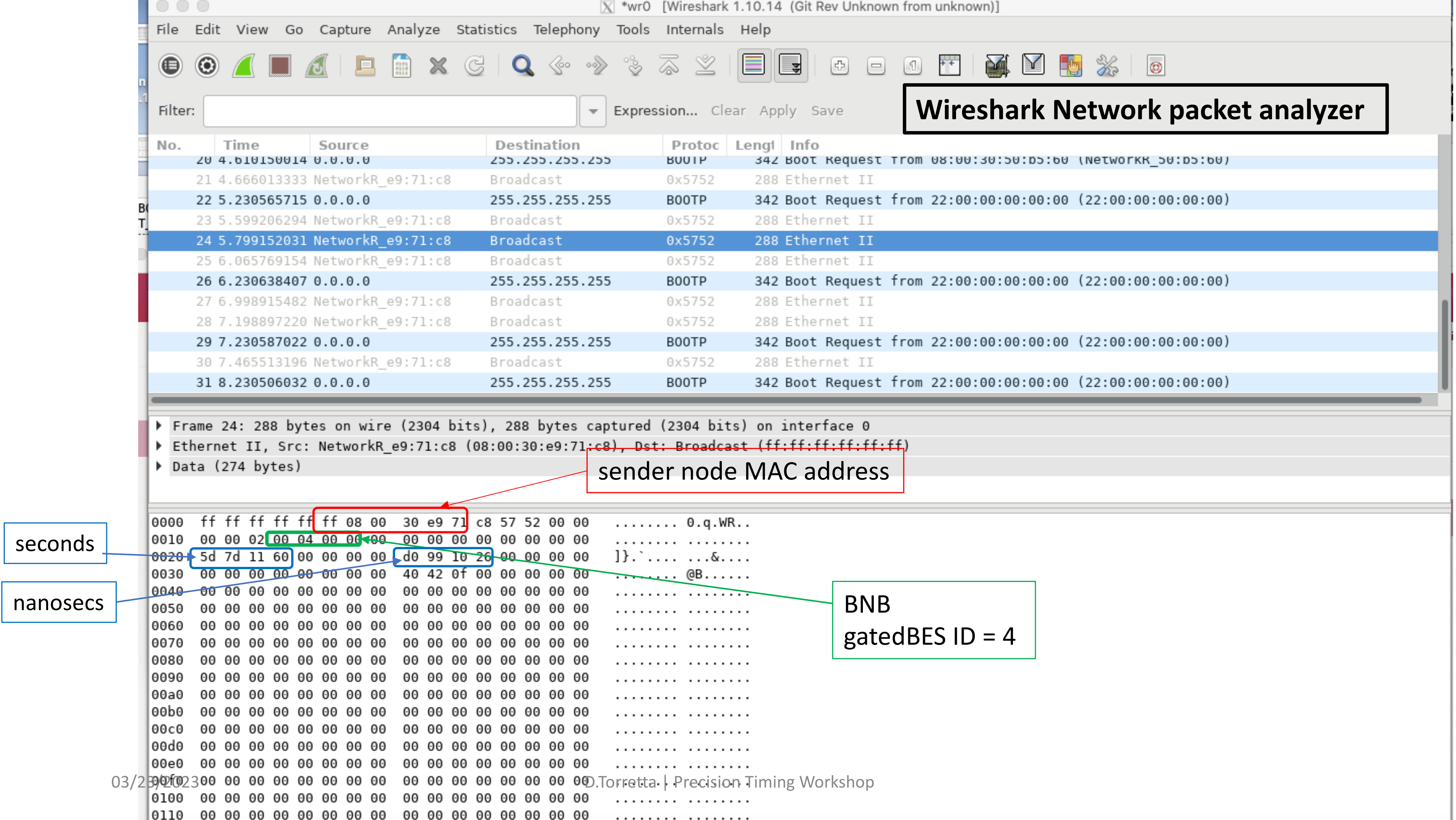
These beam signals are INPUT to the DIO mezzanine channels

The beam type ID (== DIO CH) was ADDED to the WR packets in the *wr-dio-ruler.c at sending time*

We implemented and still use the WR starting-kit software (from CERN) v2.0

```
/* first possible  location : WE USE THIS ONE */
if(inch == 1) f.cmd.value = 1;  /* $1D at MI-12;   $AE at MI-60 */
if(inch == 2) f.cmd.value = 2;
if(inch == 3) f.cmd.value = 3;
if(inch == 4) f.cmd.value = 4;  /* gatedBES at MI-12;  MIBS $74 at MI-60  */
/*printf(" f.cmd.value = %i ", f.cmd.value);*/
```

The SPEXI card in the NI trigger crate decodes the WR packets and opens the readout gates according to the beam type and origin (checking the sender node's MAC in the WR packet)


*wr0 [Wireshark 1.10.14 (Git Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
Wireshark Network packet analyzer
No. Time Source Destination Protoc Lengt Info
20 4.610150014 0.0.0.0 255.255.255.255 BOOTP 342 Boot Request from 08:00:30:50:b5:60 (NetworkR_50:b5:60)
21 4.666013333 NetworkR_e9:71:c8 Broadcast 0x5752 288 Ethernet II
22 5.230565715 0.0.0.0 255.255.255.255 BOOTP 342 Boot Request from 22:00:00:00:00:00 (22:00:00:00:00:00)
23 5.599206294 NetworkR_e9:71:c8 Broadcast 0x5752 288 Ethernet II
24 5.799152031 NetworkR_e9:71:c8 Broadcast 0x5752 288 Ethernet II
25 6.065769154 NetworkR_e9:71:c8 Broadcast 0x5752 288 Ethernet II
26 6.230638407 0.0.0.0 255.255.255.255 BOOTP 342 Boot Request from 22:00:00:00:00:00 (22:00:00:00:00:00)
27 6.998915482 NetworkR_e9:71:c8 Broadcast 0x5752 288 Ethernet II
28 7.198897220 NetworkR_e9:71:c8 Broadcast 0x5752 288 Ethernet II
29 7.230587022 0.0.0.0 255.255.255.255 BOOTP 342 Boot Request from 22:00:00:00:00:00 (22:00:00:00:00:00)
30 7.465513196 NetworkR_e9:71:c8 Broadcast 0x5752 288 Ethernet II
31 8.230506032 0.0.0.0 255.255.255.255 BOOTP 342 Boot Request from 22:00:00:00:00:00 (22:00:00:00:00:00)
Frame 24: 288 bytes on wire (2304 bits), 288 bytes captured (2304 bits) on interface 0
Ethernet II, Src: NetworkR_e9:71:c8 (08:00:30:e9:71:c8), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Data (274 bytes)
sender node MAC address
0000 ff ff ff ff ff ff 08 00 30 e9 71 c8 57 52 00 00 .... 0.q.WR..
0010 00 00 02 00 04 00 00 00 00 00 00 00 00 00 00 00 ....
0020 5d 7d 11 60 00 00 00 00 d0 99 10 26 00 00 00 00]}.`.... ...&....
0030 00 00 00 00 00 00 00 00 40 42 0f 00 00 00 00 00 .... @B......
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
00b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
00c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
00d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
00e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
00f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
0100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
0110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
seconds
nanosecs
BNB
gatedBES ID = 4

WR implementation in ICARUS

The WR switch

10 MHz & PPS form GPS on ground floor

MINOS repeater

spare

production

The WR Switch (in Grand Master mode) installed in the Timing rack in the West mezzanine floor

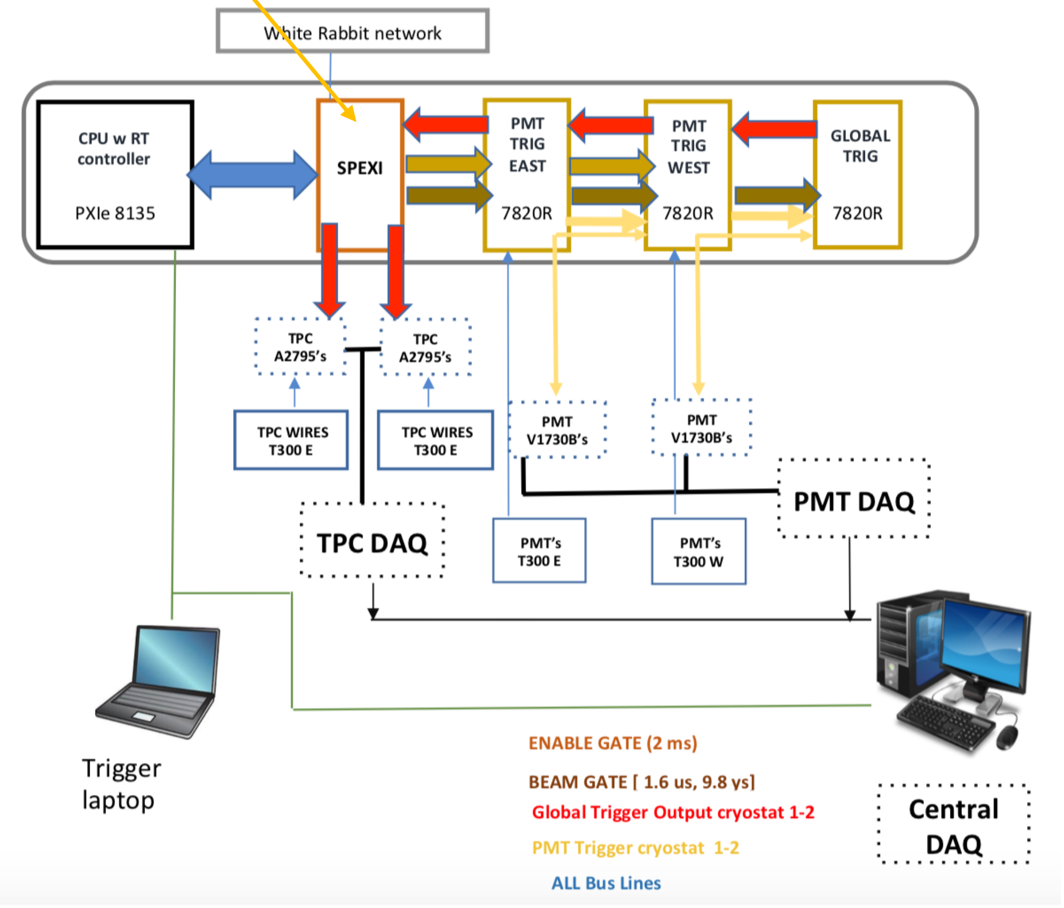
Fibers **from** MI-12 & MI-60 (beams EW inputs)

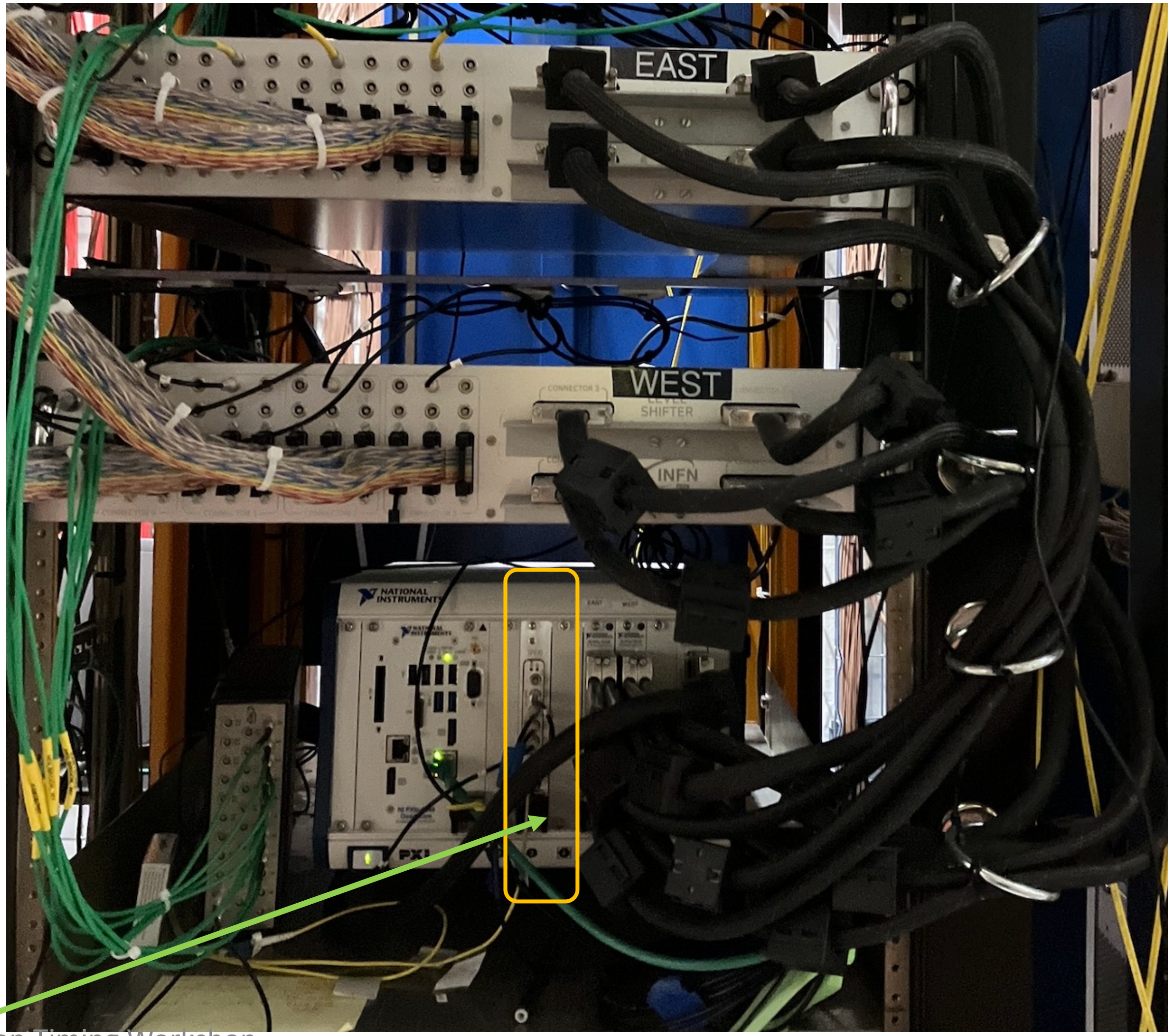
Fibers **to** the WR nodes: SPEXI and monitoring server(s)

TRIGGER SYSTEM hardware installation [see backup slides for principles overview]

The Trigger system layout is based on NI (National Instruments) PXIe instrumentation fully contained in a single PXIe crate (NI-1082) and consists of: a Real Time (RT) controller (NI PXIe-8840), **one SPEXI board** by INCAA Computers, and three FPGA programmable boards (NI PXIe-7820R).

In real life





WR fiber in DIO SPF

SPEXI functionalities

The SPEXI firmware (written by G. Meng (PD)) provides the following functionalities

- implementation of the White Rabbit protocol (WR PTP Core V4.2, released Dec 2017, last available).
- the synchronization of TPC and PMT and CRT data by providing clock and PPS (pulse per second).
- receives and decodes the beam Early Warning (absolute time) that the accelerator associates with the beam spill (BNB and NuMI).
- generates a gate signal with the resolution of 8 ns when the beam arrives. In trigger logic, the trigger is generated when the PMT signal is present in the beam gate.
- generates a enable signal, centered on the expected time extraction, for activating PMT readout.
- generates the different type of gate signals, like offbeam, MinBias and calibration, for cosmic data taking and background analysis.
- controls the synchronous trigger (via TTLink) of all the TPC mini-crates serving the TPC wires planes with independent signals for the West and East modules.
- provides info to the DAQ containing the information about beam gate and trigger conditions

Some of those signals are produces on the PXIe bus lines of the NI crate and thus seen by the other modules in the crate
And used to implement the trigger logic by the LabVIEW code.
The following 5 outputs are available from the front panel of DIO mezzanine card:

dio_o (0) : PPS output
dio_o (1) : TTLink output for TPC 1
dio_o (2) : TTLink output for TPC 2
dio_o (3) : clock (62.5MHz) output for PMT
dio_o (4) : beam gate signal output -- for test

The SPEXI initialization, as well as readout of trigger information, is performed via the trigger LabVIEW projects.

Monitoring of the beam EW signals

- As we just saw, the ICARUS triggers ***DEPENDS*** on receiving and decoding the beam EW signals by the SPEXI card.
- To ***MONITOR*** the presence and the rate of the EW we rely on software installed and running on the SPEC/DIO card mounted in a dedicated Linux server (*icarus-daq02.fnal.gov)*
- To configure the SPEC/DIO card on icarus monitoring server (and on the servers at the beam locations) we use the original WR starting-kit software and tools (from CERN) . The software is now available in this GitHub repository *https://github.com/SBNSoftware/sbndaq-wr*
  - Users need to compile the code, load the obtained kernel modules into the kernel, load drivers and gateware (that come with the product) into the hardware
- Detailed information on the above plus installation, hardware, signal distribution setup, cards initialization and troubleshooting is available here https://cdcvs.fnal.gov/redmine/projects/icarus-operations/wiki/White_Rabbit_system_Wiki
- Once the SPEC/DIO is initialized and synchronized to the WRS, you can start monitor the WR network traffic with some provided commands in the ~*tools* directory, like the one shown in the next slide.

Manual checks on beam signals (on WR nodes)

- Log into *sbn-mi12 (sbn-mi60)(*sender nodes*)* and type

```
[root@sbn-mi12 tools]# ./wr-dio-cmd wr0 stamp
ch 0, 1613419331.000000000
ch 0, 1613419332.000000000
ch 0, 1613419333.000000000
ch 0, 1613419334.000000000
ch 2, 1613419330.955842232
ch 2, 1613419331.022498408
ch 2, 1613419331.089153720
ch 2, 1613419332.155658760
ch 2, 1613419332.222315128
ch 2, 1613419332.288971384
ch 2, 1613419333.355482264
ch 2, 1613419333.422138480
ch 2, 1613419333.488794840
ch 3, 1613419330.920476432
ch 3, 1613419330.987133000
ch 3, 1613419331.053788160
ch 3, 1613419332.120293008
ch 3, 1613419332.186949568
ch 3, 1613419332.253605936
ch 3, 1613419333.320116872
ch 3, 1613419333.386773040
ch 3, 1613419333.453429208
ch 3, 1613419334.519945448
[root@sbn-mi12 tools]#
```

PPS on CH 0

Local CH 2 (gatedBES)

Local CH 3 ($1D)

- Log into *icarus-daq02* (destination node) and type

```
[root@icarus-daq02 tools]#
[root@icarus-daq02 tools]# ./wr-dio-cmd wr0 stamp
ch 0, 1613419413.000000000
ch 0, 1613419414.000000000
ch 0, 1613419415.000000000
ch 0, 1613419416.000000000
ch 1, 1613419412.558627048
ch 1, 1613419412.625290912
ch 1, 1613419412.691956480
ch 1, 1613419412.758622640
ch 1, 1613419413.758608008
ch 1, 1613419413.825273584
ch 1, 1613419413.891937848
ch 1, 1613419413.958602112
ch 1, 1613419414.958586376
ch 1, 1613419415.025250840
ch 1, 1613419415.091915416
ch 1, 1613419415.158580776
ch 1, 1613419416.158563048
ch 1, 1613419416.225228520
ch 1, 1613419416.291893688
ch 1, 1613419416.358558952
ch 4, 1613419412.559827192
ch 4, 1613419412.626491720
ch 4, 1613419412.693156928
ch 4, 1613419412.759823064
ch 4, 1613419413.759808856
ch 4, 1613419413.826474056
ch 4, 1613419413.893138392
ch 4, 1613419413.959802728
```

PPS on CH 0

Remote CH 1 ($1D)

Remote CH 4 (gatedBES)

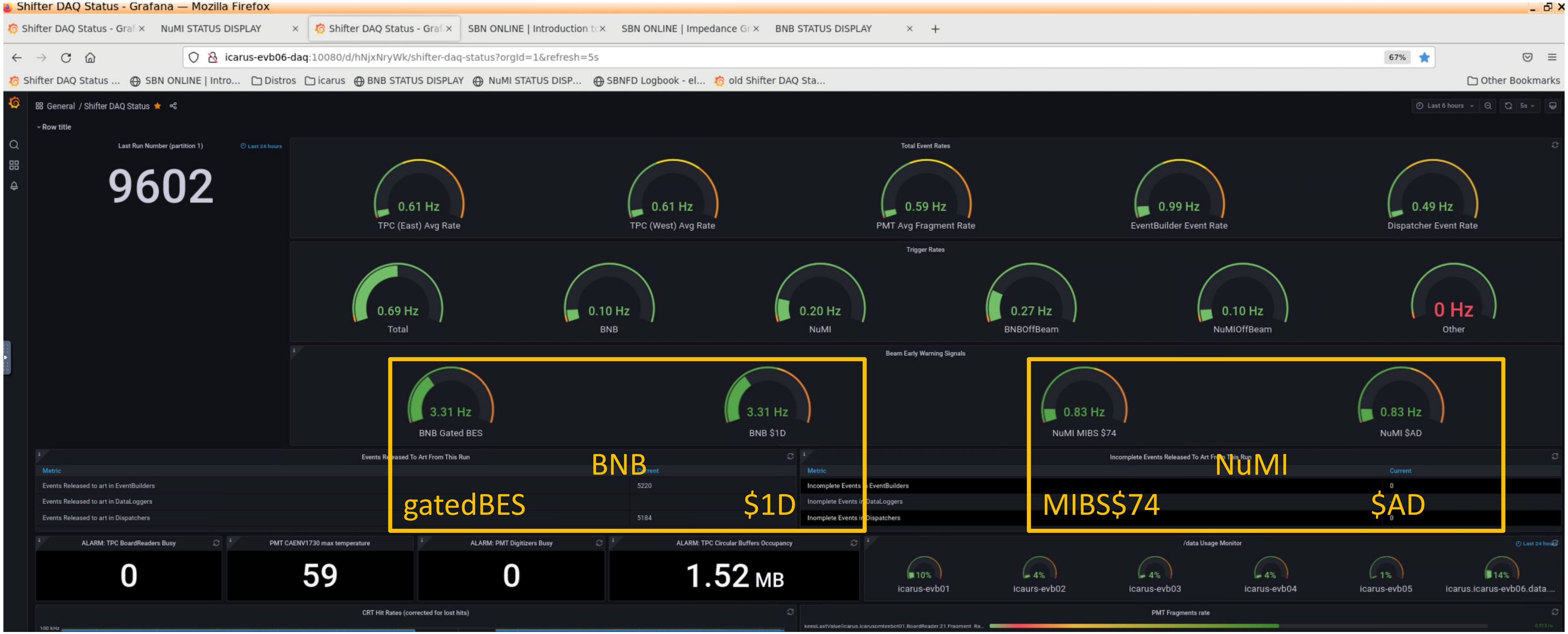
Need of a continuous monitor of the beam signals: enters the *WR boardReader*

- A software module in the ICARUS daq that receives the WR network packets containing the beam signals information and *continuously* monitors the beam rates while taking data
- Original code written by Bill Badgett (thanks!), modified by D. Torretta to add EW beam signals metrics to Grafana (next slide)
- The WR boardReader runs on *a WR node (server) equipped with a SPEC/DIO card synchronized to the whole WR network*
- It gets the EW beam signals timestamps at start run from channels [0-4] in the WR DIO mezzanine card
    - CH 0 is the WR PPS; Ch 1-4 are the EW
- Defines rates metrics [4] to Grafana which displays the rates of the four EW beam signals

DAQ rates (GRAFANA) display

Rates are not always 100% accurate but are an indication of EW signals coming to FD.

IF ZERO, and the IFBEAM pages instead show EW being issued, something is wrong in the EW signals distribution



General comments on setup, operations and troubleshooting the WR network/hardware

- We routinely use the procedures and tools provided by the CERN WR starting-kit software to initialize, reset, synchronize the WR hardware we use , in particular the SPEC/DIO cards
  - We find that when all works , the system does not require any further intervention (self calibrating, accuracy …)
  - But at times it take time to fix issues (cards not synchronizing, no PPS,…)
- We have implemented and still using version 2.0 (2014) of the software
  - It’s getting old and newer versions are available, v3.0 and v3.1 but they require a change of the OS from SLF7 to Ubuntu LTS 16.04 and 18.04.
  - Tests were performed on a dedicate server.
- Hardware issues: some hardware (SPEC) is not available anymore.
  - we procured a few spares but we may want to look into new equipment: which one? And what support will be provided by vendor or CERN ?

Thanks for your attention