KM3NeT
Italy

Report from the
TriDAS
Working Group

Tommaso Chiarusi

INFN - Sezione di Bologna

INFN sez Bologna
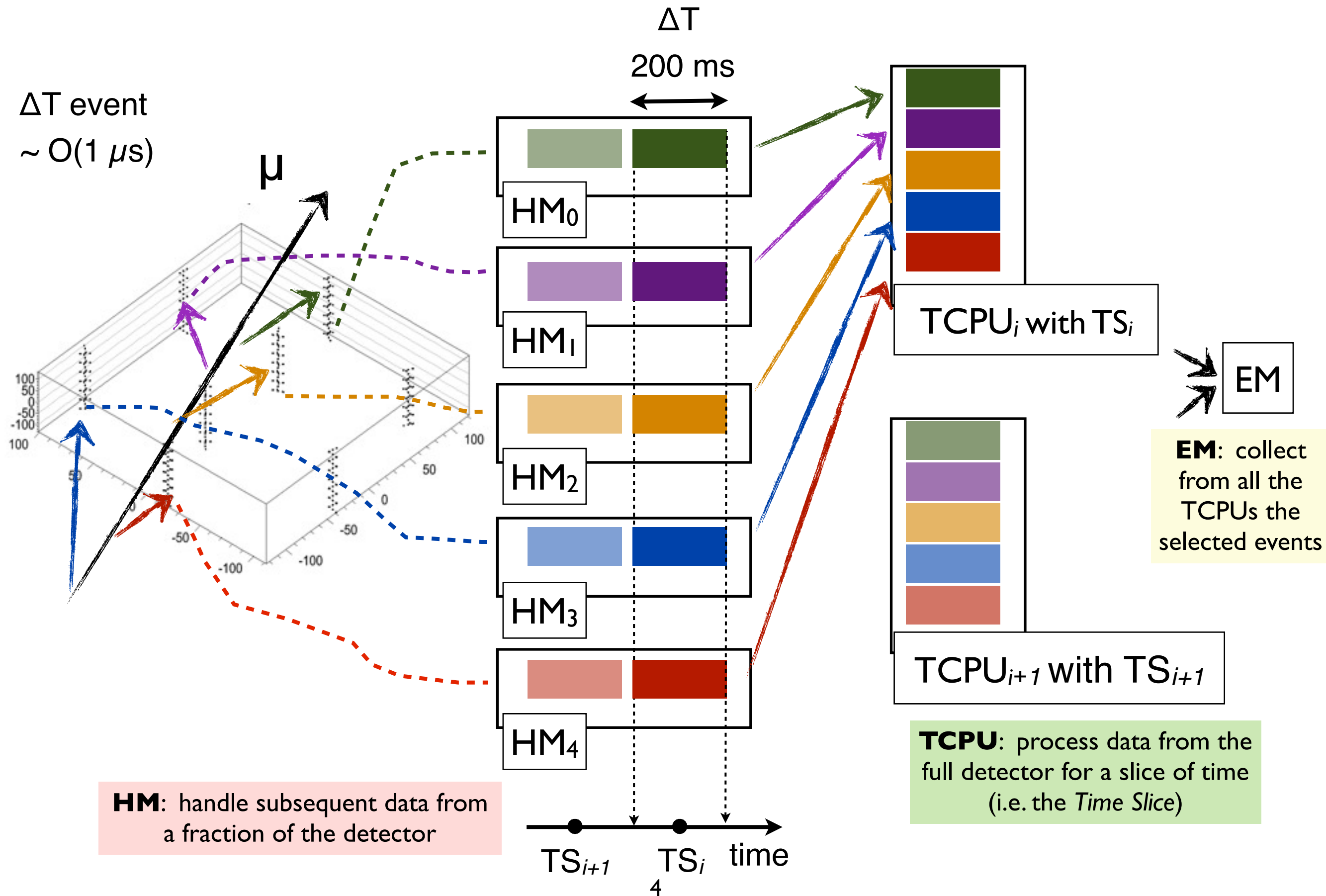
1

Summary of the talk

- The refurbished TriDAS crew

- New TriDAS for the 8 Towers: what is kept...

- ... and what is changed!

- tests with the NEMO F2 Tower

- same Trigger process for on && off line

- test-bench in Bologna

- integration Basic-TriDAS-kit

- Validation HW

- Planning & Milestones

TriDAS crew

Dept.	name	position	% time (*)	topic
● Bologna	Tommaso Chiarusi	staff	**100%**	coordination
Bologna	Carmelo Pellegrino	1.5y INFN fellow - 2.5y Ph.D.	**100%**	software devel.
○ Bologna	Andrea Paolucci	staff	20%	system manager
Bologna	Luigi Antonio Fusco	1.5 Ph.D student	10%	s/w devel for TCPU - offline
CNAF	Matteo Manzali	? INFN fellows- 1.5y Ph.D.	**80%**	software devel.
CNAF	Francesco Giacomini	staff	**>30%**	design and review
CNAF	Stefano Zani	staff	20%	network design
LNS-CNAF	Lucia Morganti **or** Matteo Favaro	2y INFN fellow	**100%**	software devel.
○ Pisa	Bachir Bohuadef	? INFN fellow	**>30%**	trigger algos
● Pisa	Giuseppe Terreni	staff	**<30%**	software devel.

(*) real fraction of time, not the officially declared FTE percentage

What is kept: the TriDAS modularity (and most of the C++ code)

ΔT event
$\sim O(1\ \mu s)$

μ

ΔT
200 ms

HM$_0$

HM$_1$

HM$_2$

HM$_3$

HM$_4$

TCPU$_i$ with TS$_i$

TCPU$_{i+1}$ with TS$_{i+1}$

EM

EM: collect from all the TCPUs the selected events

TCPU: process data from the full detector for a slice of time (i.e. the *Time Slice*)

HM: handle subsequent data from a fraction of the detector

TS$_{i+1}$ TS$_i$ time

4

What is changed 1/5: use of (more) robust software for some items

M. Manzali, F. Giacomini, C. Pellegrino, T. Chiarusi

CMake — context for software compilation, building, and deployment

boost C++ LIBRARIES — threads, templates (containers) C++11 and beyond

ØMQ — inter-server/ -process/ -threads communications (will replace ControlHost)

git **Atlassian Bitbucket** — repository/version maintainer (substitute CVS) + web interface (tracker)

Jenkins — continuous integration, test runner, static analysis, package compiler

See M. Manzali's poster @ CCR 2014 (prize for best poster!!!)

http://wiki.infn.it/_media/cn/csn2/km3/km3_ccr_poster_2014_05_27_matteo.pdf

About code repository: the ownership policy

**git** **Atlassian Bitbucket** https://bitbucket.org/chiarusi/tridas

if code is private (current status): restricted number of developers

if code is open-source: unlimited number of developers

I put the request to the SC to be authorized to make open-source our codes (as CMS's and ATLAS's, for example!). Naturally inserting always the INFN copyright.

What is changed 2/5: TSC as Supervisor (renamed as TriDAS Supervising Center)

| all HMs | TSC | TCPUj |

available j

Decide TS α

TS: α to TCPU j

sending TS: α to TCPU j
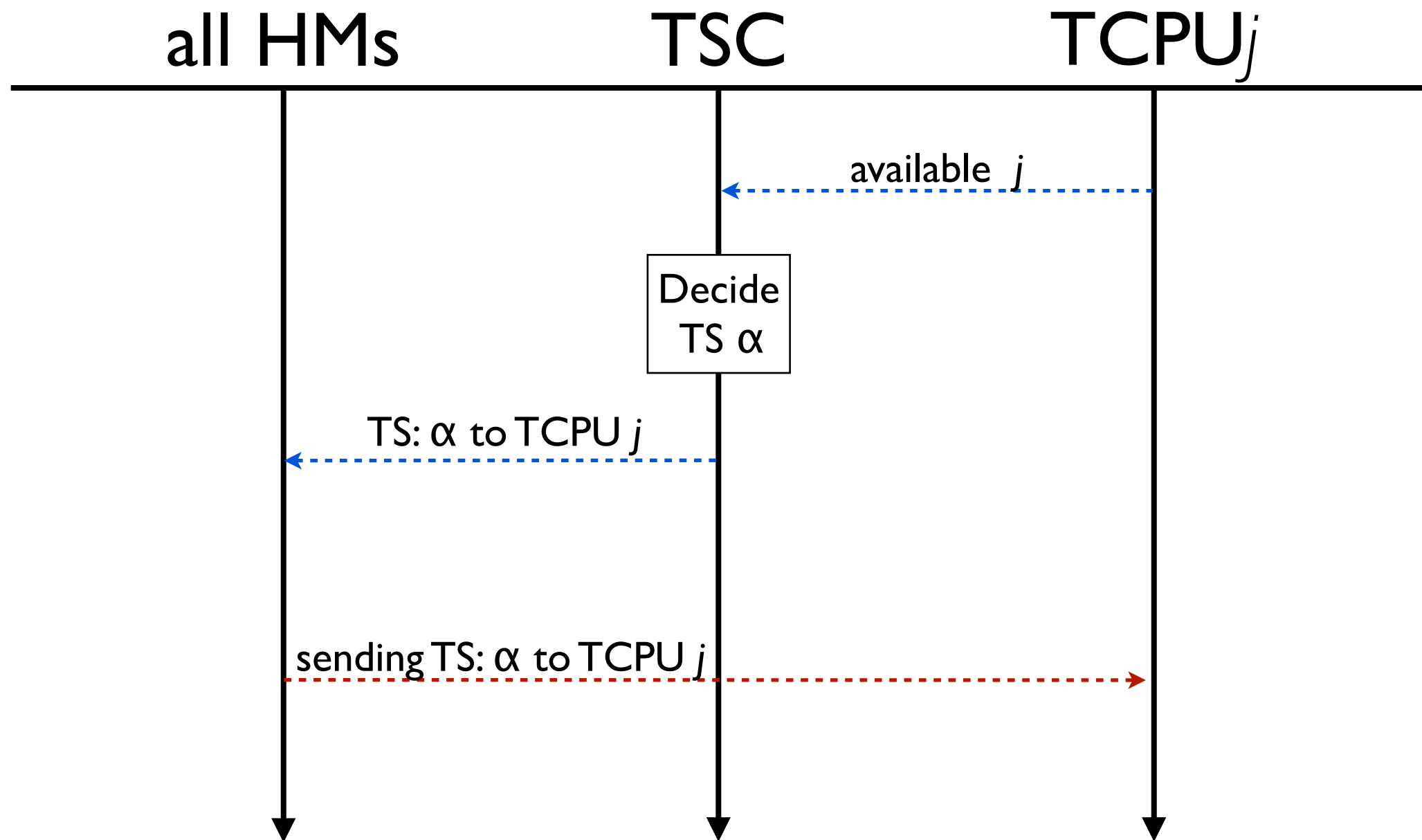
Note: with Phase 3, absolute time embedded in rawdata.
This implies the TSC to _decide when_ starting the run(see next slide)

What is changed 3/5: the start of a Run

On start-up, the HMs automatically slices the time by knowing the *timeslice duration* (200 ms) and by referring to the *KM3-origin-of-time* (e.g. 01-01-2015 @ 0:0:0 ????). *This means that the timeslice numbering is unique* (e.g. it will exist one and only one TS #1, and it is that one starting on 01-01-2015 @ 0:0:0. - TS #2 is that one starting 200 ms after TS #1)

The HMs don't send the timeslices to the TCPUs if not commanded by the TSC

The TSC shares the common DAQ GPS time synchronization, and translates the time info in TS ID. So, to a Run-start time it will correspond one and only one TS ID.

E.g. : The TriDAS is active since 01-01-2015 @ 0:0:0, but the Run is started only on 02-01-2015 @ 0:0:0 (one day later):
so the first TS ID of the run is 432000 (= 86400 s * 5 TS /s)

Using an unsigned int (32 bit) for the TS ID we cover ~27 years!

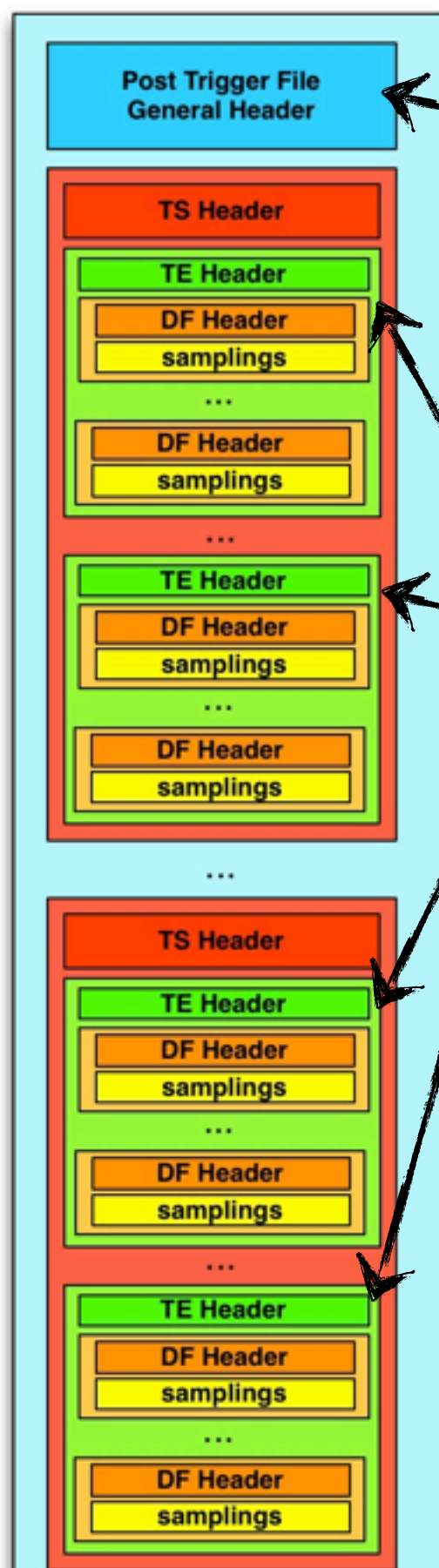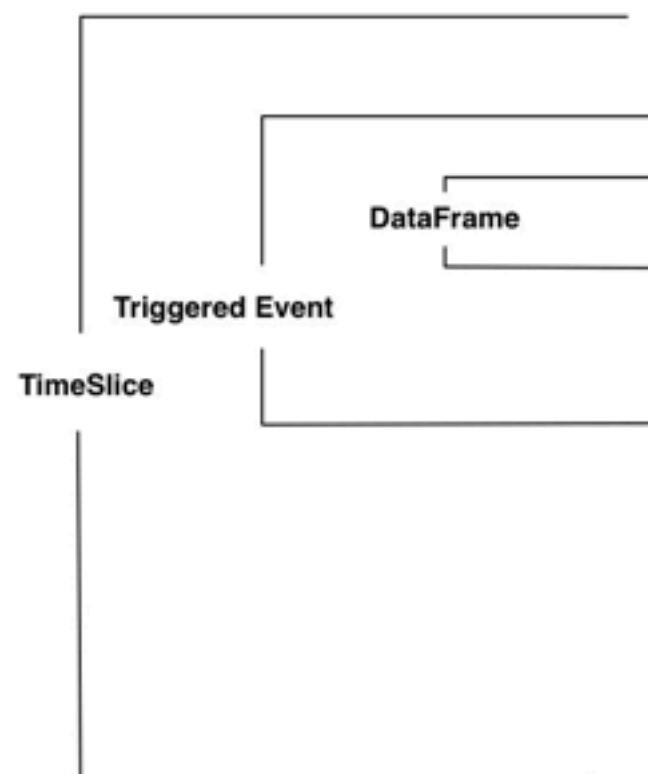# What is changed 4/5: TSC as interface to the Console and DB Interface

C. Bozza, T. Chiarusi, A. Rovelli

DB LNS ⟷ DB Napoli

Remote locations

Shore Station in Portopalo

Upload logs and get updated run-type configs

Logs on files

DB Interface – Cache manager

TSC

TriDAS

Configuration for run-types

Get run-types

TriDAS GUI

Console

Data Manager

commands and acks

storage
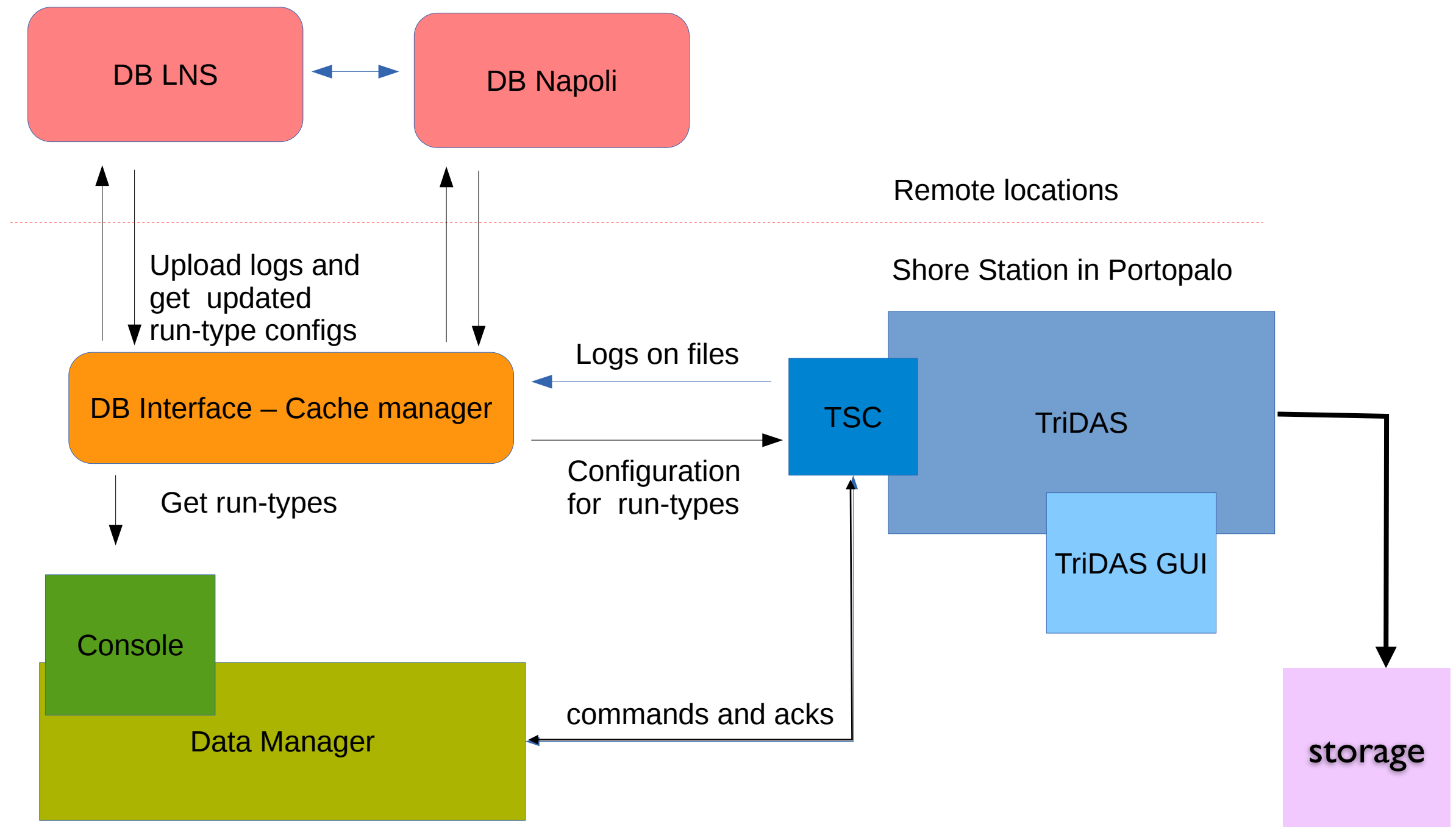
see Cristiano Bozza's talk

9

T. Capone, T. Chiarusi, R. Coniglione,
C. Distefano, M. Manzali

1. Name of the PTfile with only a sequential ID number.

2. Added the runsetup DB code (corresponding to the detailed list of all the trigger algos and their parameterizations)

3. No more wrap info: *we have absolute time*

After the TriggeredEvent (TE) header and *before* the raw-data excerpt, there will be a **list with the hits contributing to the trigger** (with a trigger seed mask)

This imply a maximum overhead of ~ 10% if (unlikely) all the hits of an event contribute to the trigger.

The new Post-Trigger format (*bit coded*) will be published on our Wiki as soon as it will be implemented

Switch to the new Raw data format *(F. Ameli & C.)*

Struttura del Data frame.
- 1   DWORD (32 bit) Header.1: Sync & ID
- 1   DWORD (32 bit) Header.2: Time
- 1   DWORD (32 bit) Header.3: Time
- 1   DWORD (32 bit) Header.4: Hit Info
- N   DWORD (32 bit) Dati: Samples

Formato Header.1:
- (31:16) Sync Bytes
- (15:9) ID Torre (0-127)
- (8:4) ID Piano (0-31)
- (3:0) ID PMT (0-15)

Formato Header.2:
- (31:0) Tempo assoluto GPS(47:16)

Formato Header.3:
- (31:16) Tempo assoluto GPS(15:0)
- (15:00) Tempo relativo in unità di 5 ns

Formato Header.4:
- (31) Flag Compressione (1 se il DF e'compresso)
- (30) Flag Frammentazione (1 se l'evento é frammentato)
- (29:8) Carica (compressa/non compressa) associata a questo Data Frame. La carica é la somma di tutti i campionamenti
- (7) FEM FIFO Full
- (6) Non usato
- (5:0) nSamples. Questo campo rappresenta il numero di campioni nelle DWORD che seguono l'header; il numero di DWORD è funzione della flag di compressione e può essere calcolato come segue:
  - Flag Compr. = 0 -> nDword = ceil(nSamples/4);
  - Flag Compr. = 1 -> nDword = ceil(nSamples/2);

Formato Dati (compressi):
- (31:24) Campionamento 1 (espresso in canali dell'ADC)
- (23:16) Campionamento 2 (espresso in canali dell'ADC)
- (15:08) Campionamento 3 (espresso in canali dell'ADC)
- (07:00) Campionamento 4 (espresso in canali dell'ADC)

Formato Dati (non compressi)
- (31:16) Campionamento 1 (espresso in canali dell'ADC)
- (15:00) Campionamento 0 (espresso in canali dell'ADC)

Ameli (spring '14)

Absolute timing !!!

Bit range	Campo	Formato	Range
47 – 40	Anni	BCD	0 - 99
39 – 30	Giorni	BCD	0 - 365
29 – 13	Secondi	Straight Binary	0 - 85399
12 – 0	Frame Counter (125 us)	Straight Binary	0 - 7999

New "FCMServer" simulator completed(C. Pellegrino et al.)

TriDAS modified accordingly (M. Manzali e C. Pellegrino)

New raw data format: new `FCMSimu`

C. Pellegrino, T. Chiarusi, F. Giacomini, M. Manzali, G. Terreni

Two programs realized:

- ## Data Converter
  Aim: converts NEMO Phase 2 FCM raw data into NEMO Phase 3 format.

- ## FCMSimu
  Aim: simulate the data stream produced by a NEMO floor.
  Features: uses real hit samples, supports fragmented hits, tunable hit rate.

Data Converter

C. Pellegrino, T. Chiarusi, F. Giacomini, M. Manzali, G. Terreni

From Phase 2 to Phase 3, some changes: mainly to TimeStamp fields. Also the position of some field has changed.

The Data Converter (aka `P2toP3Converter`) produces a hit file in the Phase 3 format from a Phase 2 **raw** data file.

It preserves the validity of all the informations contained in the input file **but with no compression on** samples: sample size is 2 Byte/sample (compression can be toggled on/off).

It is useful for simulating data stream (see FCMSimu, next slide) in the Phase 3 format.

Phase 2
Hit file

010110
110011
101000
0001

Phase 3
Hit file

010110
110011
101000
0001

13
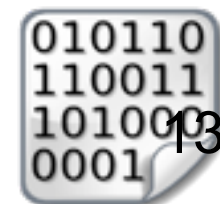
New FCMSimu flowchart

C. Pellegrino, T. Chiarusi, F. Giacomini, M. Manzali, G. Terreni
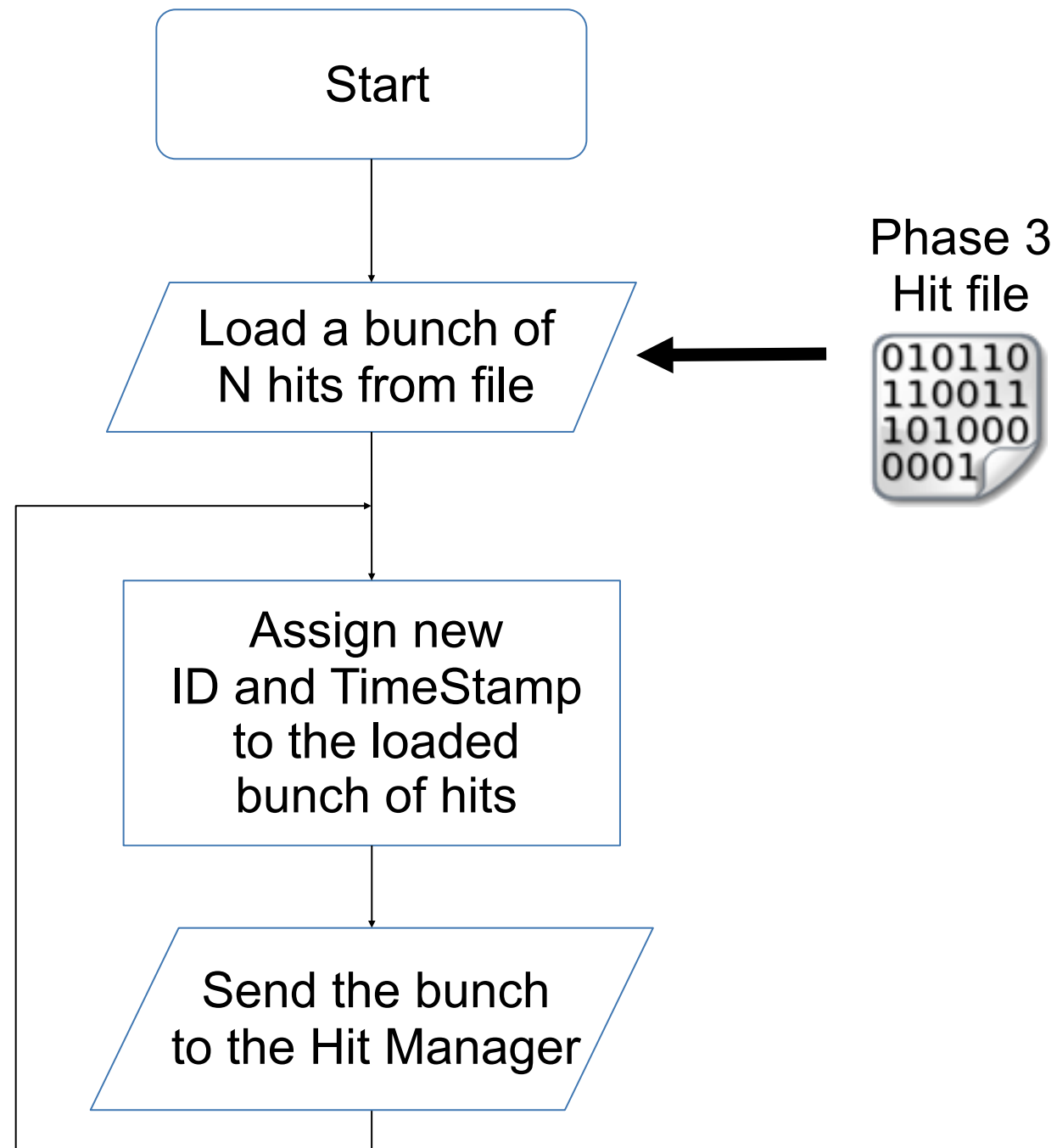
The **time of arrival** of a hit for a given PMT is determined by summing to the time of the previus hit for that very PMT a randomly generated time ΔT, which follows the exponential distribution:

$$\lambda\, e^{-\lambda \times \Delta T}\,,$$
$$\lambda = Hit\ Rate$$

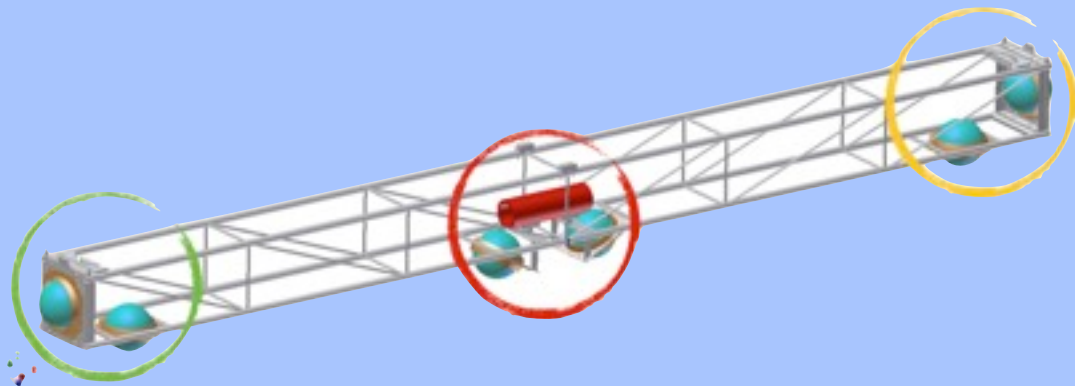At each update of the TimeStamp, the ID informations are re-assigned rotating the PMTID in the [0-5] range.

Trigger Levels

All the time-sorted hits of a TS

time

L1

L1 event duration
(6μs)

L2
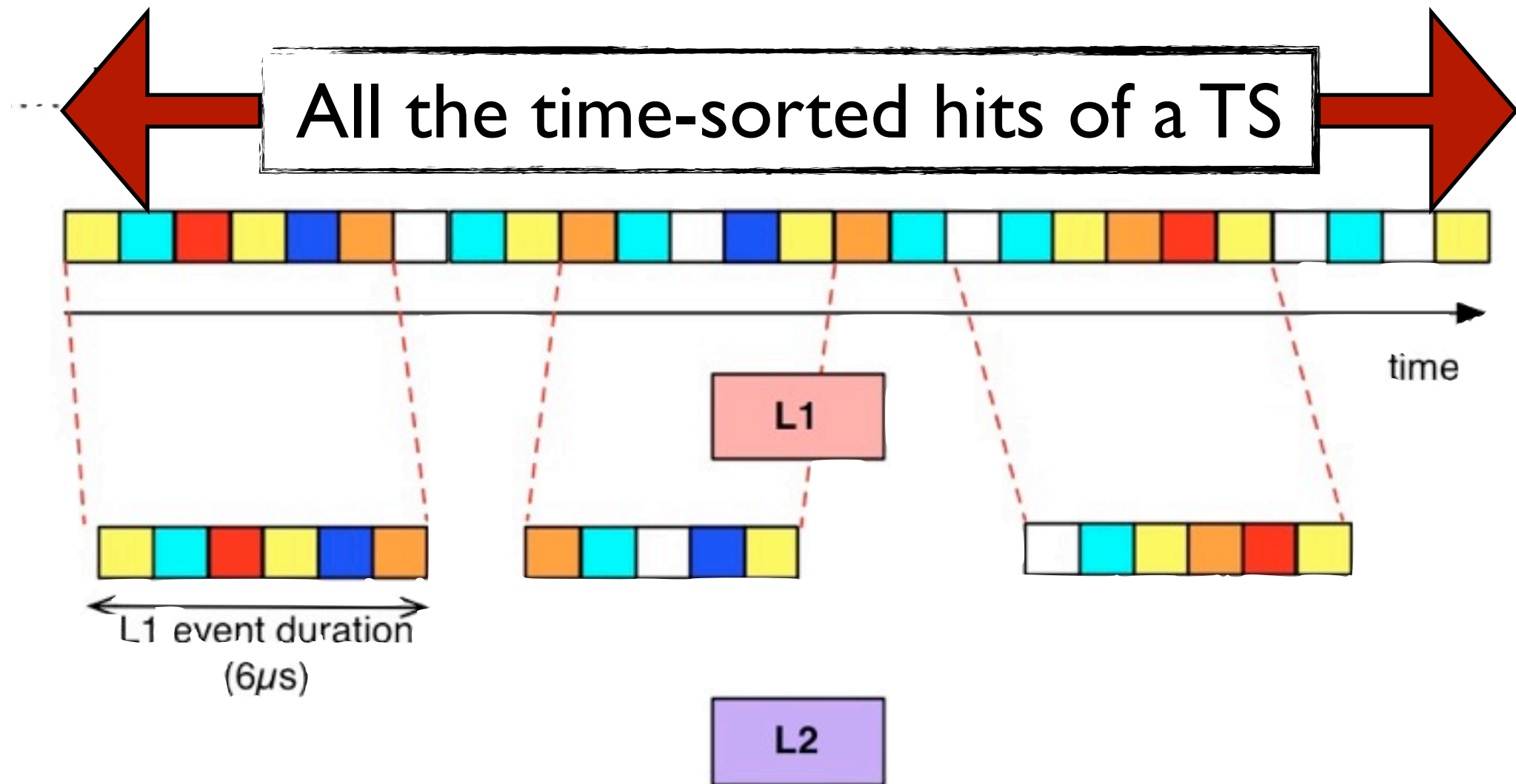
L1 = PRESELECTION

L2 = Dedicated Trigger Algorithm

A — Simple Coincidences
$\Delta T \leq 20$ ns
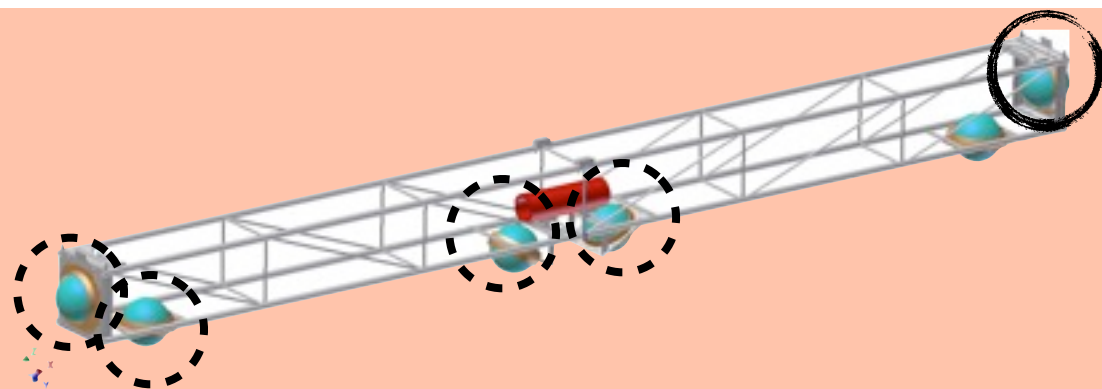
B — Floor Coincidences
$\Delta T \leq 80/40$ ns

C — Shooting hit
$Q > Q_{th}$
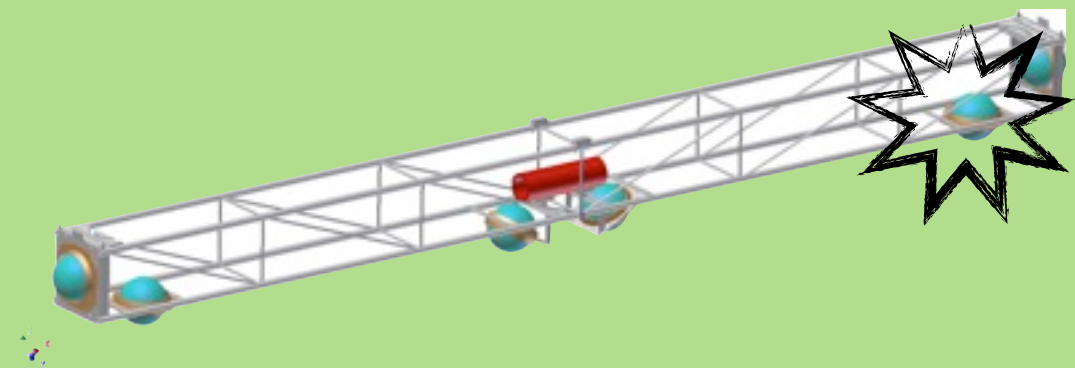
Sampling Window ΔT

$$\nu_{out} = \frac{1}{\Delta T}\binom{n}{k}p^k(1-p)^{n-k}, \qquad p = 1 - e^{-\Delta T\,\nu_{bkg}}$$

n = n. involved PMTs

k = minimum searched hits within ΔT

L2 = Dedicated Trigger Algorithm

For a proposal of various trigger algorithms, please refer to my presentation at Coll. Meeting in November.
(https://agenda.infn.it/getFile.py/access?contribId=14&sessionId=2&resId=0&materialId=slides&confId=7015)

At present: ScanDir algo implemented (see internal note)
(http://wiki.infn.it/_media/cn/csn2/km3/tcpu_performances.pdf)

Trigger Simulation/Validation Task Force

Objective: define a strategy to validate the proposed trigger algorithms producing figure of merits (efficiency, purity etc.).

Involved people

TriDAS WG   Annarita   Carla, Rosa e Piera

Additional help is more than welcomed!

Use of the TriDAS for Offline Trigger application
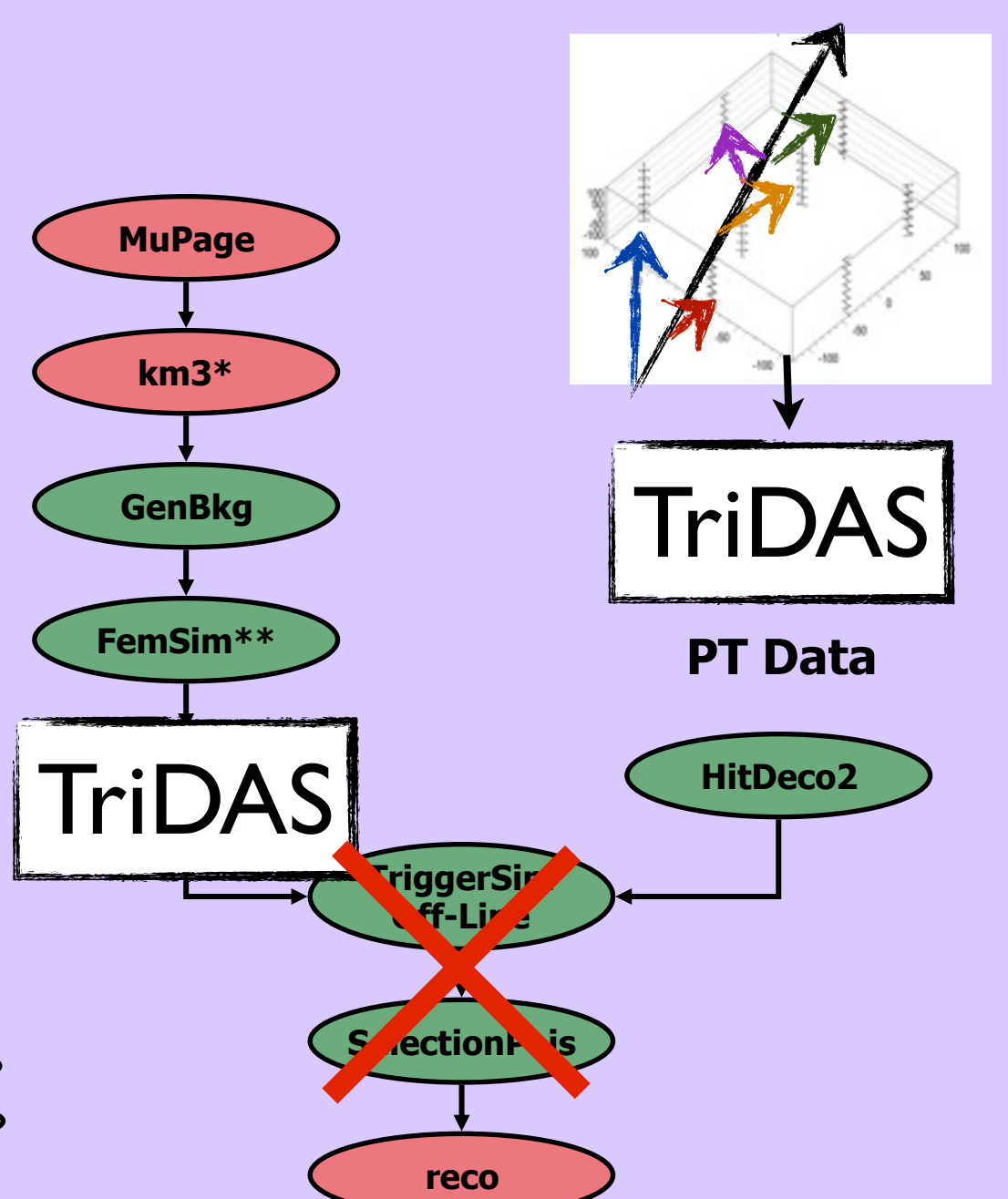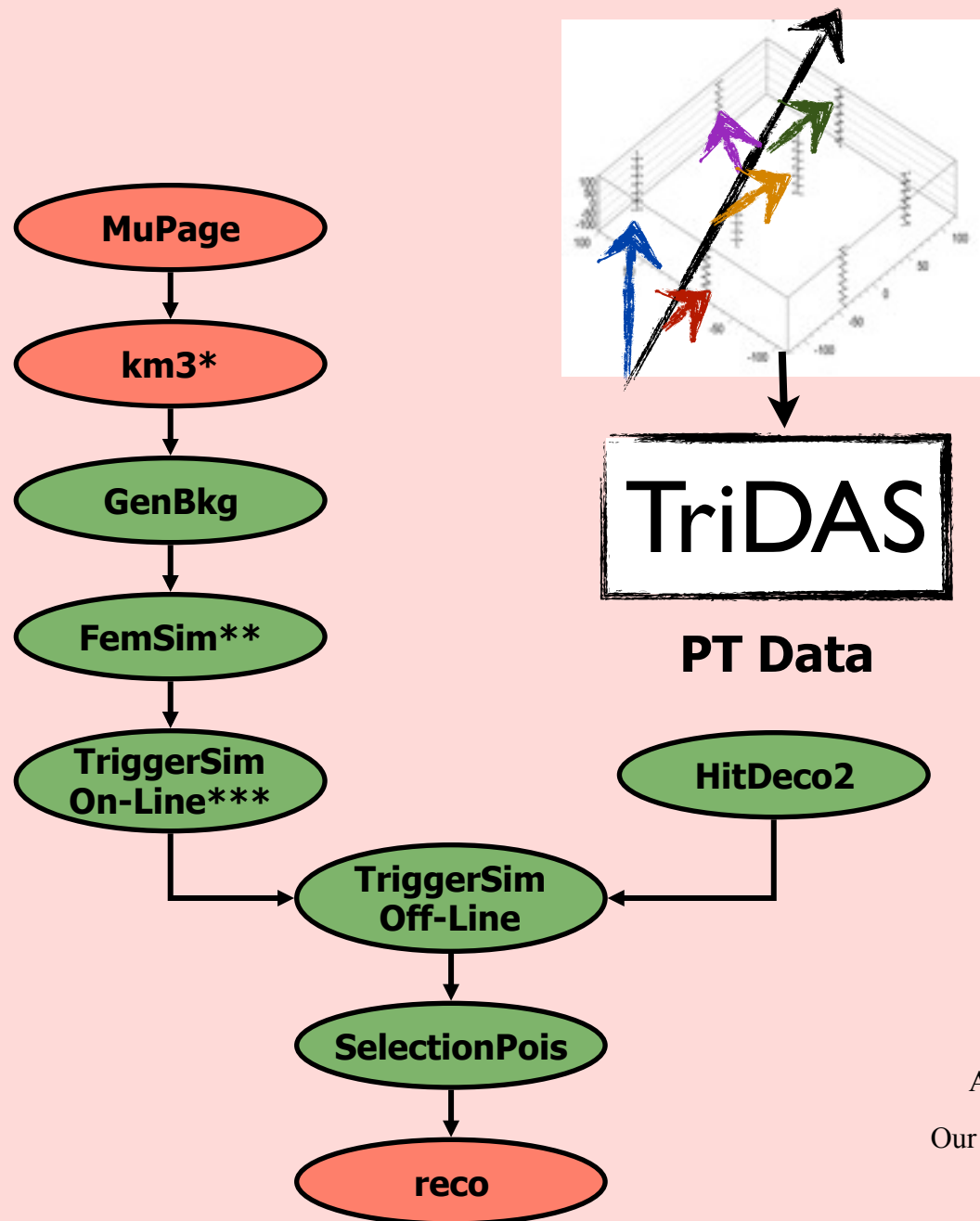(not simulation nor reproduction: it is the very online trigger applied to MC events)

TCPU-ONLINE

Post FEMSim ascii file

TCPU-OFFLINE

each evt is treated as a Time Slice

Only one HM_th
STSReceiver overloaded

MQ_STSFull

TTS_th
TTSAssembler

MQ_STSFree

FREE_th
Releaser

MQ_TTSFree

MQ_TTSReady

MQ_TTSDone

Plugin_trigger

EM_th
TTSManager overloaded

MQ_TTS2EM

do_trigger broadcast signal

MQ_TriggerDone

MQ_TEFree

TRIGGER_th
TriggerInterface

Write to file

TCPU Offline : Status

L.A. Fusco, T. Chiarusi, M. Manzali, C. Pellegrino
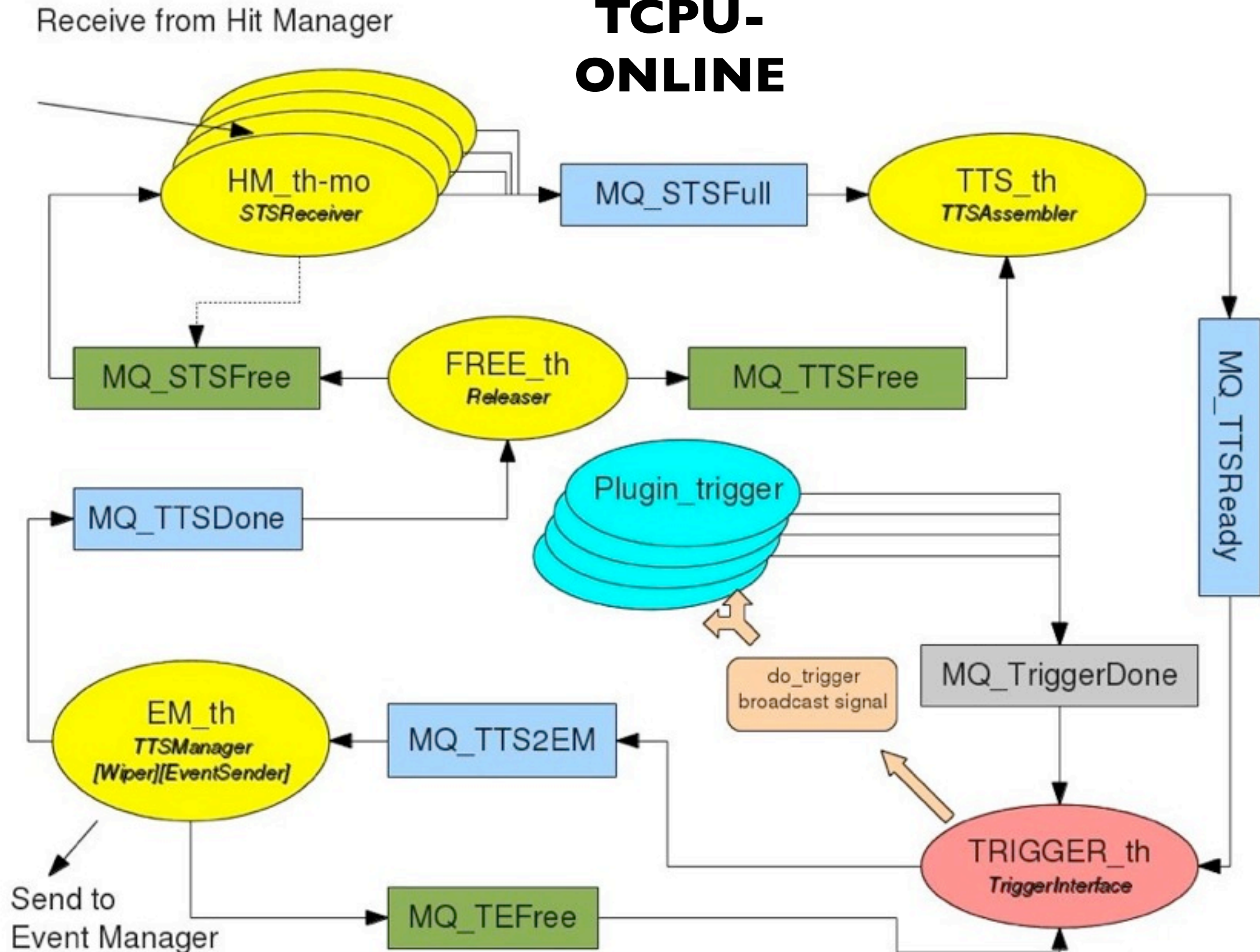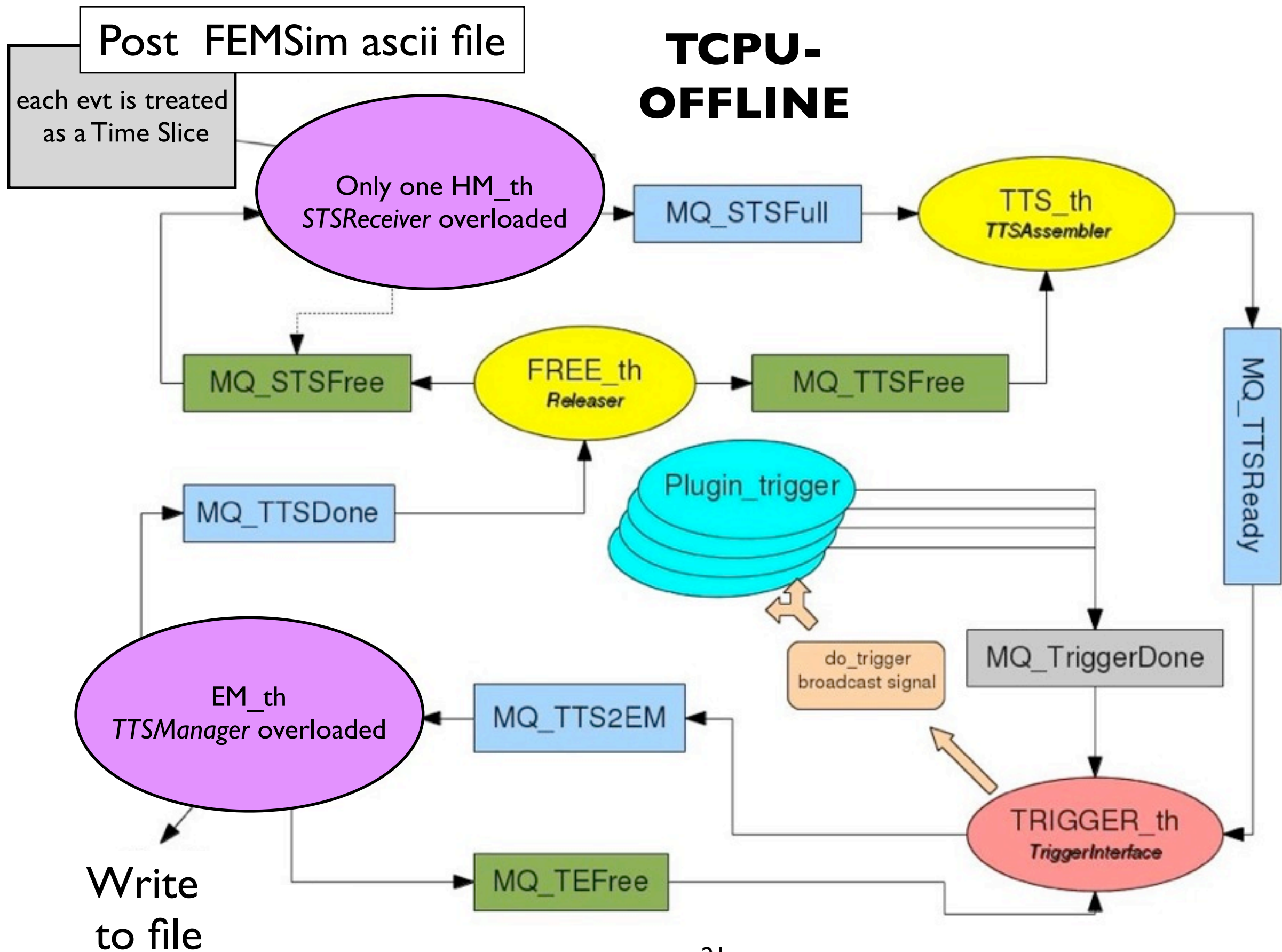
- **Create STSs from simulated data:**
- Then the TCPU will act as if they came from the towers.
- Datacard for offline;

- **STSReceiver modified to deal with simulated (after FemSim) data:**
- ReceiveSTS() method → reproduce the creation of the STS by the HitManager using the already allocated sts_;

  - FemSim data parsing;
  - DataFrame creation;
  - sts_ filled with the read data (dataframes, samples, sts header etc.).

- Completed STSs are passed to the proper queue, then go through the usual TCPU processes.

TCPU Offline : TO-DO list

L.A. Fusco, T. Chiarusi, M. Manzali, C. Pellegrino

- **Modify the code according to the Phase3 data format:**
– Read properly the FemSim file;
– Use the updated dataformat.

- **Produce a readable (.evt?) output:**
– Decide what to write in the .evt files.

- **Tests on simulations.**

First version finalized by the end of September
(Luigi is then temporary leaving for ECAP)

Test with the NEMO F2 Tower

Since April, we made several tests of the new TriDAS (released after a major revision) with the Nemo F2 Tower.

The purpose was to verify the consistency of the new processes with respect to the "Classic" ones.
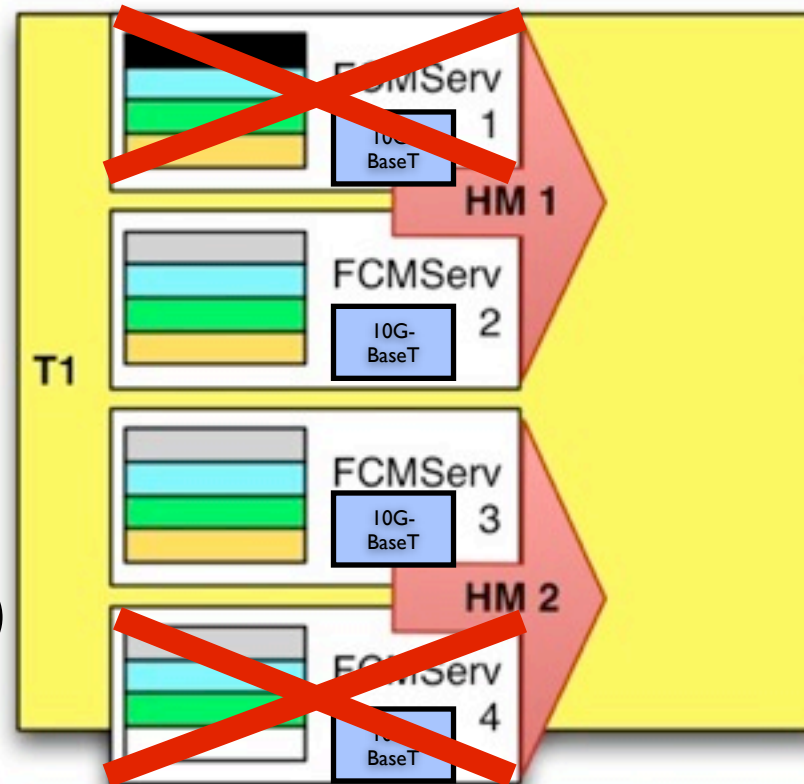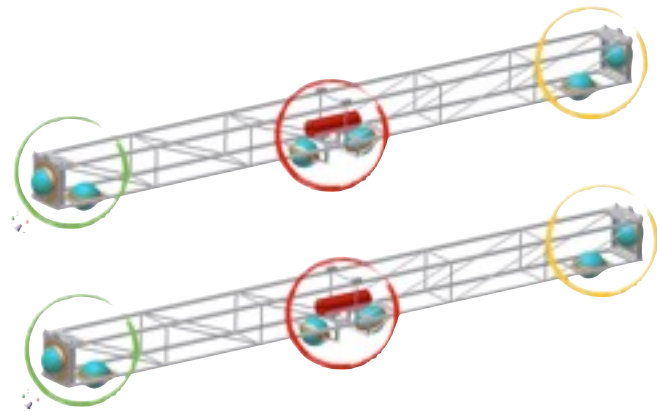
The tests were generally succesfull:
- Hit Manager .... ok
- TriggerCPU .... ok
- Event Manager .... under completion

The new TriDAS has a backward compatibility with Phase 2. Tests will last till the Tower shutdown.
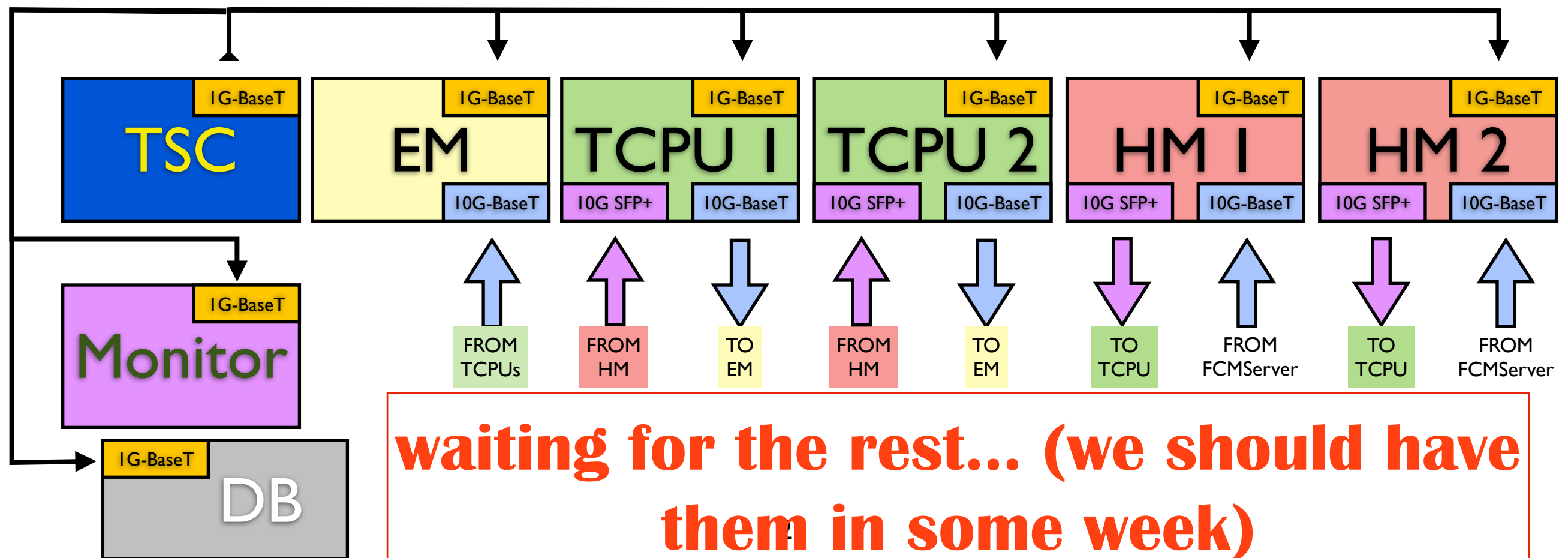
Test Bench in Bologna/Tower integration

2 FCMServers+ 2 Terasic

Already in Bo

8-12 FEM
2-4 FCM
2 FCMServer (to Rome Group)
1 TwinSquare (2HM+2TCPU)

| TSC | EM | TCPU 1 | TCPU 2 | HM 1 | HM 2 |

FROM TCPUs | FROM HM | TO EM | FROM HM | TO EM | TO TCPU | FROM FCMServer | TO TCPU | FROM FCMServer

Monitor

DB

waiting for the rest... (we should have them in some week)

Basic-TriDAS-kit

1. **Basic TriDAS for Integration site**: the Basic-TriDAS-kit offers services (it is not validated in the integration site: that is done at the Bologna Test-Bench)
     a) dump rawdata;
     b) file-post trigger L1 (SC/FC, charge overthershold trigger seed)
     c) online monitoring of the sensible observable (customizing *for any need...*)

2. **IMPORTANT**: the Basic-TriDAS-kit must be tested @ Bologna test bench. Some of the machine must be send from the LNS. Unfortunately we got delays because the already purchased servers had wrong NICs (error of the vendor). Replacement done. Validation done

       2 FCMServers to be addressed to Roma
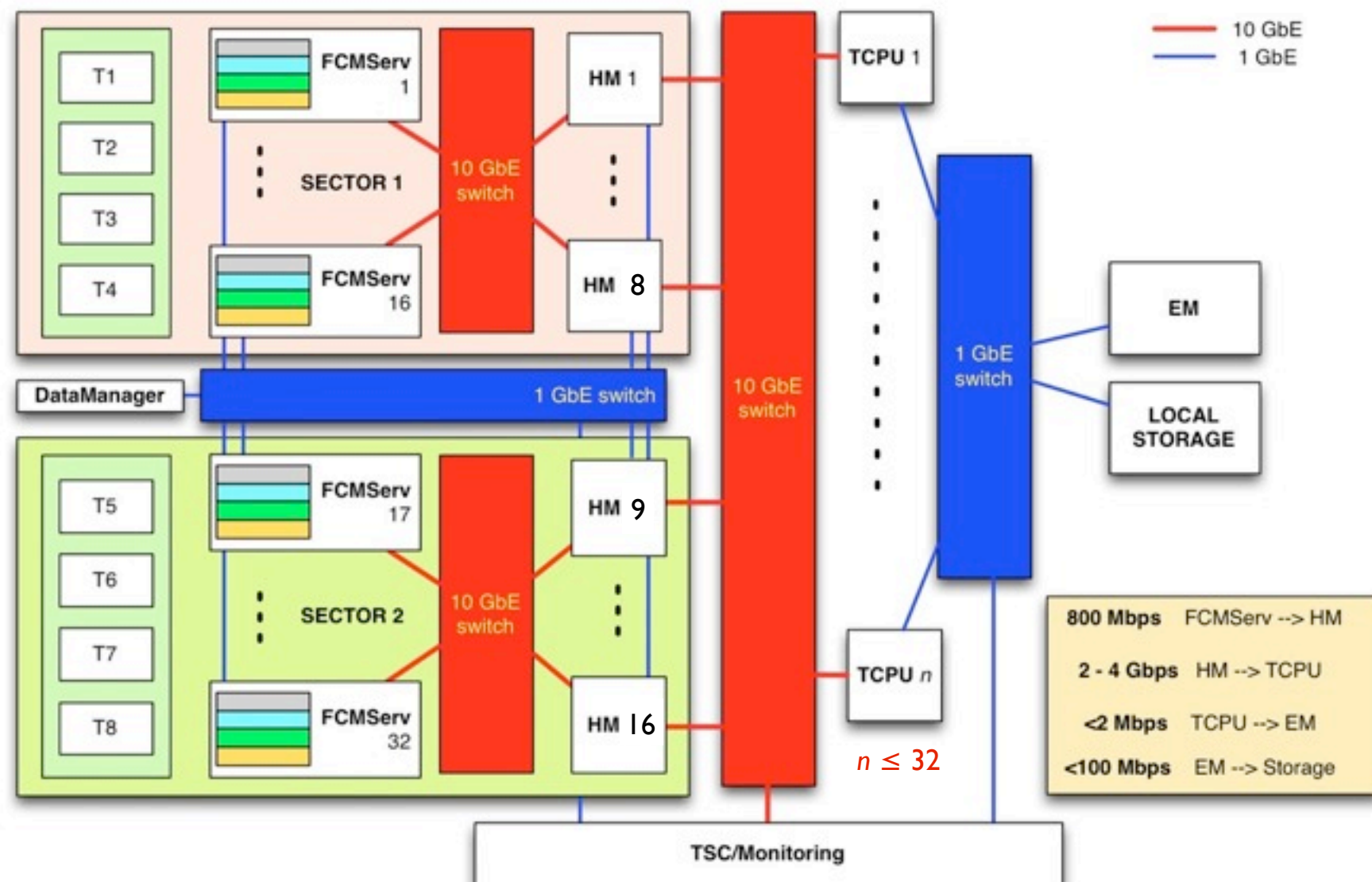   2 FCMServers + 1 TwinSquare server to be addressed to Bologna.
       Which is their current status?

3. **Planning**: **to be confirmed (depending when we receive the servers).**

4. **Requests**:
     a) Each integration expert tell if TriDAS is useful or useless for his/her test.
     b) External access layout (VPN, VNC) : *who did overcome M. Imbesi ?*

Validation and installation of TriDAS shore station

1. more than 80 nodes
2. 10 GbE switches
3. cablings

800 Mbps FCMServ --> HM

2 - 4 Gbps HM --> TCPU

<2 Mbps TCPU --> EM

<100 Mbps EM --> Storage

n ≤ 32

* servers validation: already defined a procedure:
- check of basic HW parts
- connectivity tests (with `iperf`)

* server installation: procedure to define together **with LNS personnel** (A. Rovelli, M. Musumeci)

*10 GbE switches: installation and validation **with VENDOR support**

| item	(principally) assigned to	status	planning	Milestones
Basic Revision of TriDAS framework	M. Manzali and C. Pellegrino	done		First Tower Integration
New raw dataformat compliance	M. Manzali and C. Pellegrino	finalized, to be tested	end of july	
New FCMSimu	C. Pellegrino	finalized, to be tested	end of july	
Test with Switch Niagara	T. Chiarusi and S. Zani	started	end of july	
Deployment software on Integration servers and tests	T. Chiarusi and A. Paolucci	not yet started	mid august	
New post-trigger data format	M. Manzali	started	end of september	
TSC as a supervisor	F. Giacomini and M. Manzali	not yet started	end of november	First Tower Deployment
TSC and DB Interface	T. Chiarusi, new fellow and C. Bozza	not yet started	end of november	
TSC and Console	T. Chiarusi, new fellow and A. Rovelli	not yet started	end of november	
New TriDAS GUI	new fellow	not yet started	end of november	
TriDAS network validation and installation	T. Chiarusi and S. Zani and DELL	not yet started	not yet defined	
Shore Station Servers validation and installation	A. Paolucci and ?? (from LNS)	not yet started	not yet defined	
TCPU offline	L.A. Fusco	started	end of september	
Trigger algorithm implementation and validation	T. Chiarusi, C. Pellegrino, M. Manzali and Trigger Task Force	started	not yet defined	

SPARES

L2 - Generalities

Summary:

a) T- trigger
b) Simple Causality trigger
c) Sky scan trigger
d) Tracking source trigger
e) Vertex-splitting trigger
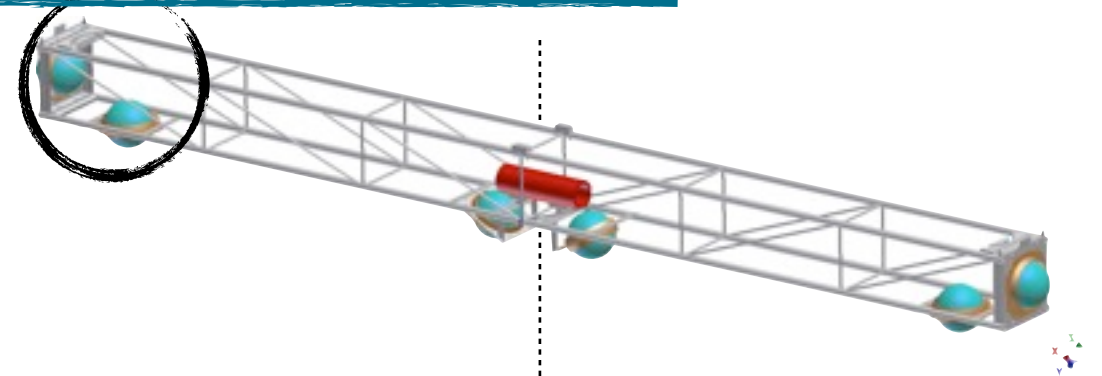f) Follow-up trigger (GRB, supernovae, GW)

The dedicated L2 algorithms process only the hits yielding the **L1s** **but for the tracking source trigger** which combines all the hits of a **L1** event
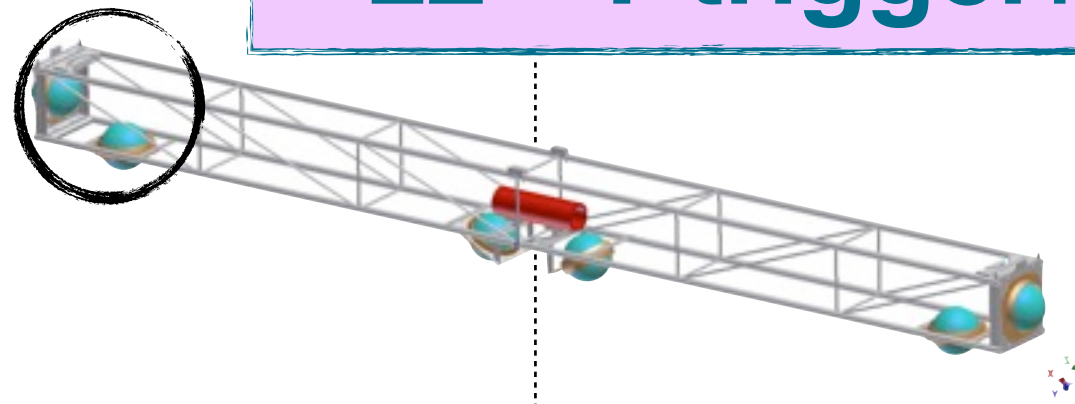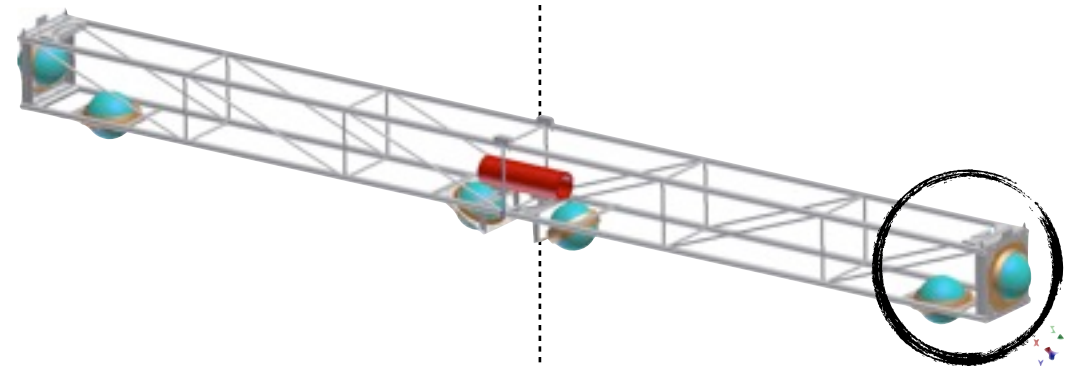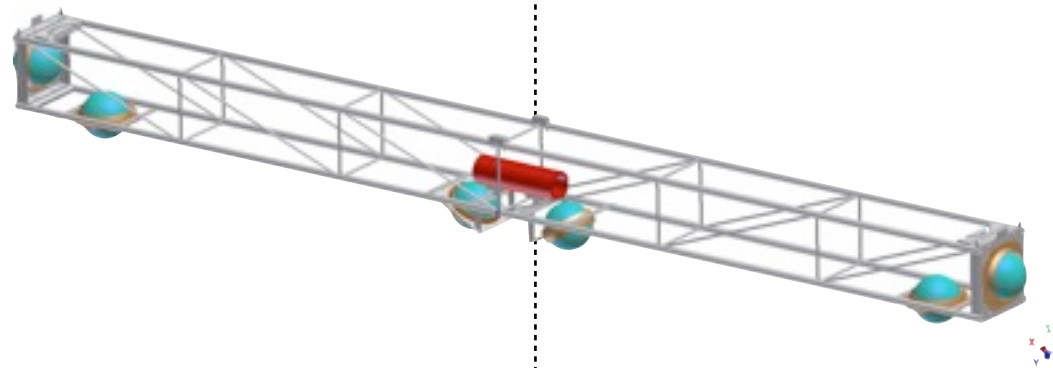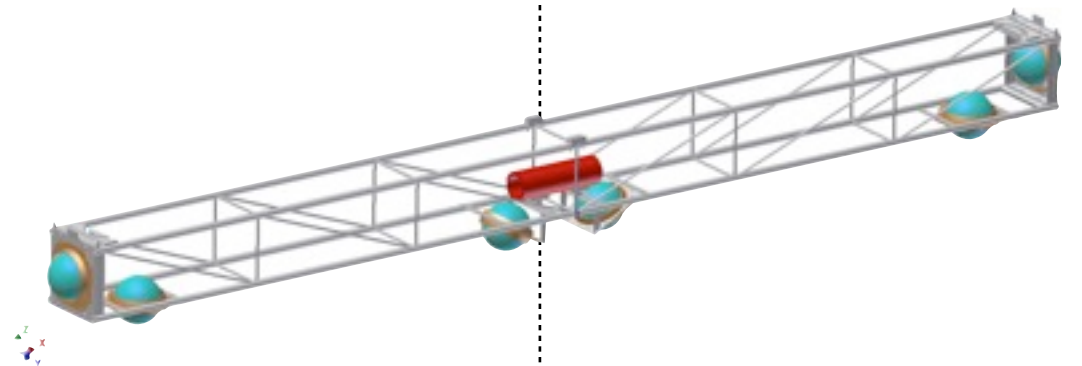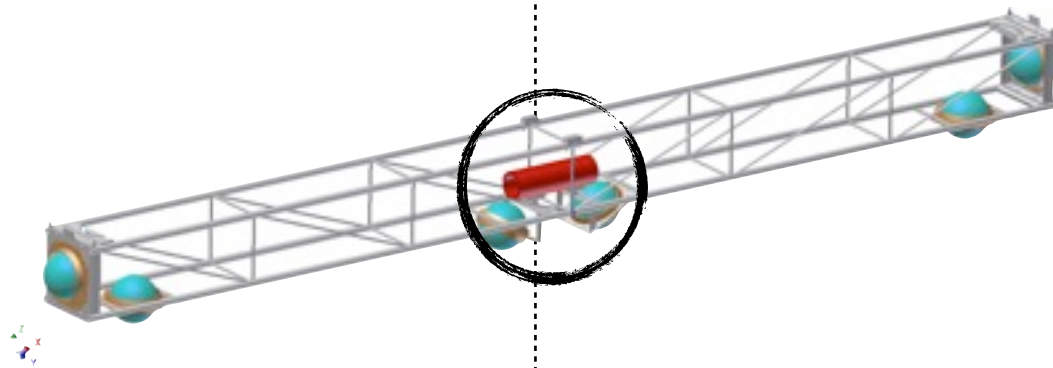
T₂

T₃

The trigger is set when the n. of consecutive T_2 or T_3 pairs is $\geq N_{th}$ within a certain time-window ΔT

L2 - **Simple causality trigger**

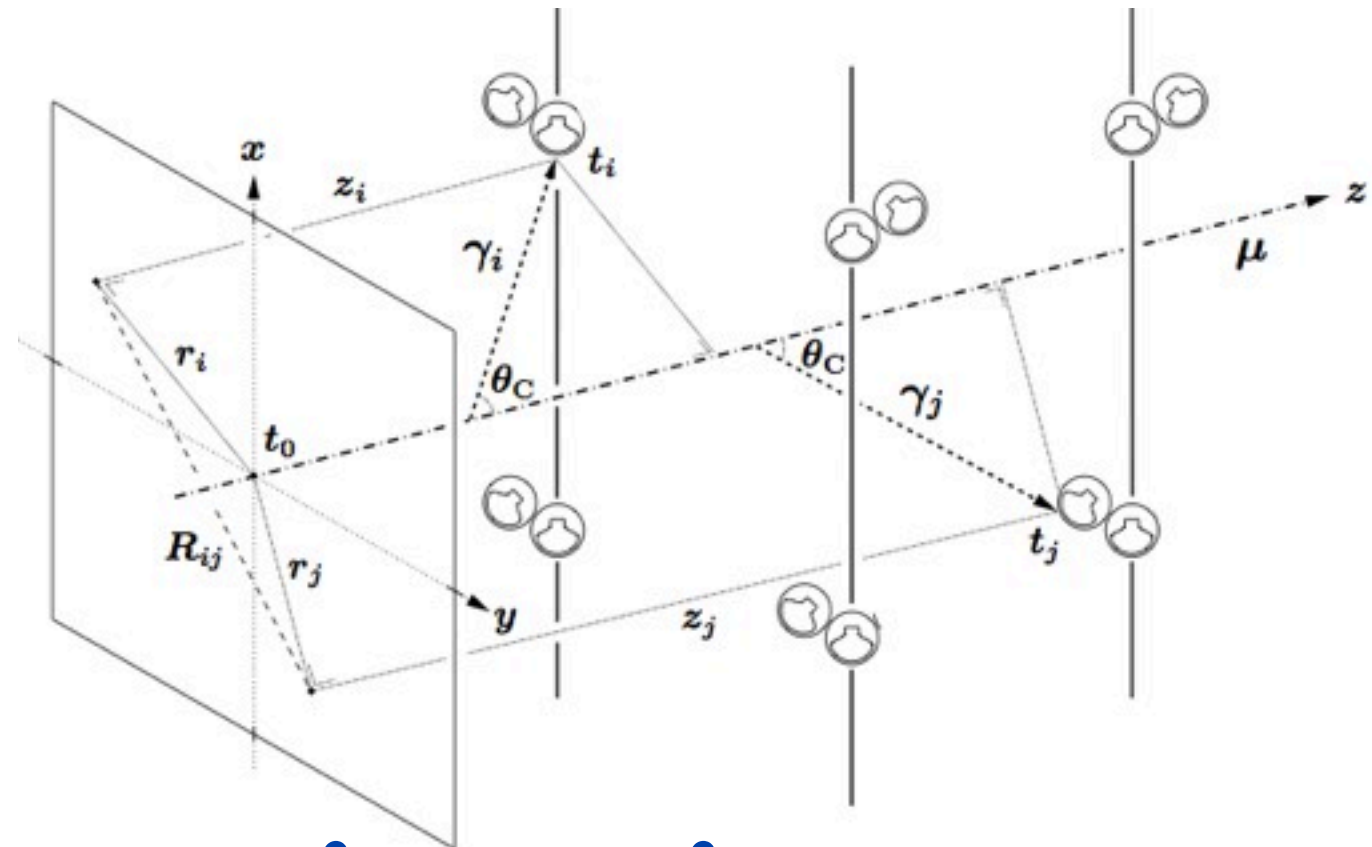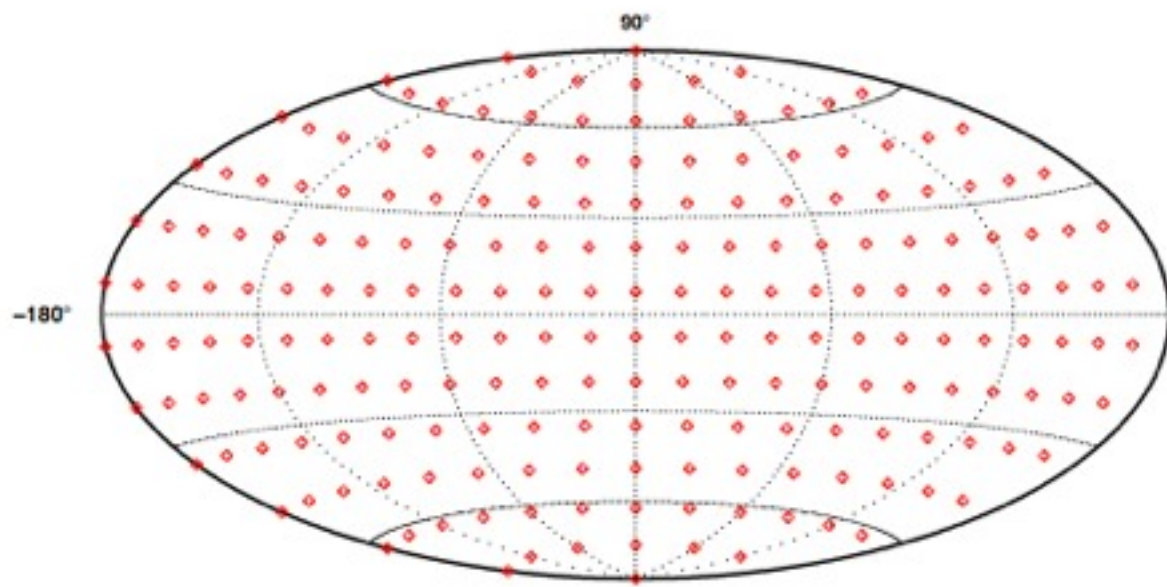1. A minimum n. of ***consecutive*** L1s $\geq$ N_{th} within a ΔT (at least $n_{PMTs} \geq 5$)

2. *3D*-causality filter : $\left| t_i - t_j \right| \leq \left| \vec{x}_i - \vec{x}_j \right| \dfrac{n}{c}$

3. The trigger is set if the n. of satisfying hits is $\geq N'_{th}$

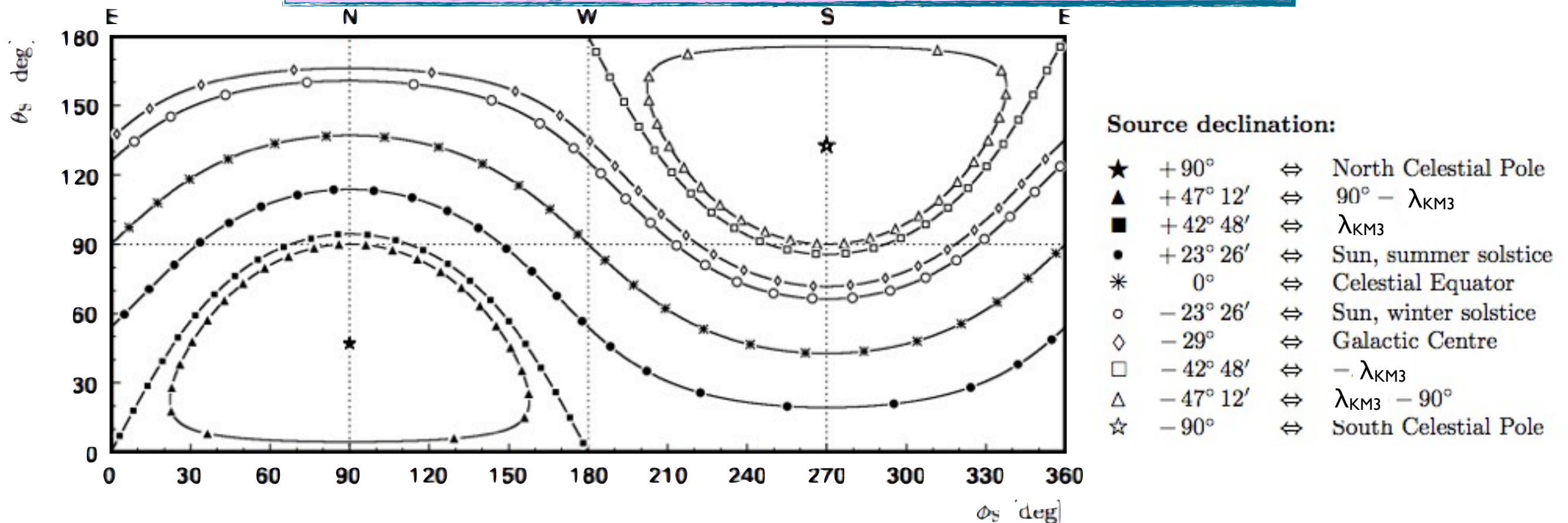1. A minimum n. of ***consecutive*** L1s ≥ N_{th} within a ΔT (at least $n_{PMTs} \geq 5$)

2. A homogeneous sky survey is done → "***rotation***" *procedure:* µ // z

$$|(t_i - t_j)c - (z_i - z_j) | \leq \tan\theta_c \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]} = \tan\theta_c | R_{ij} |$$
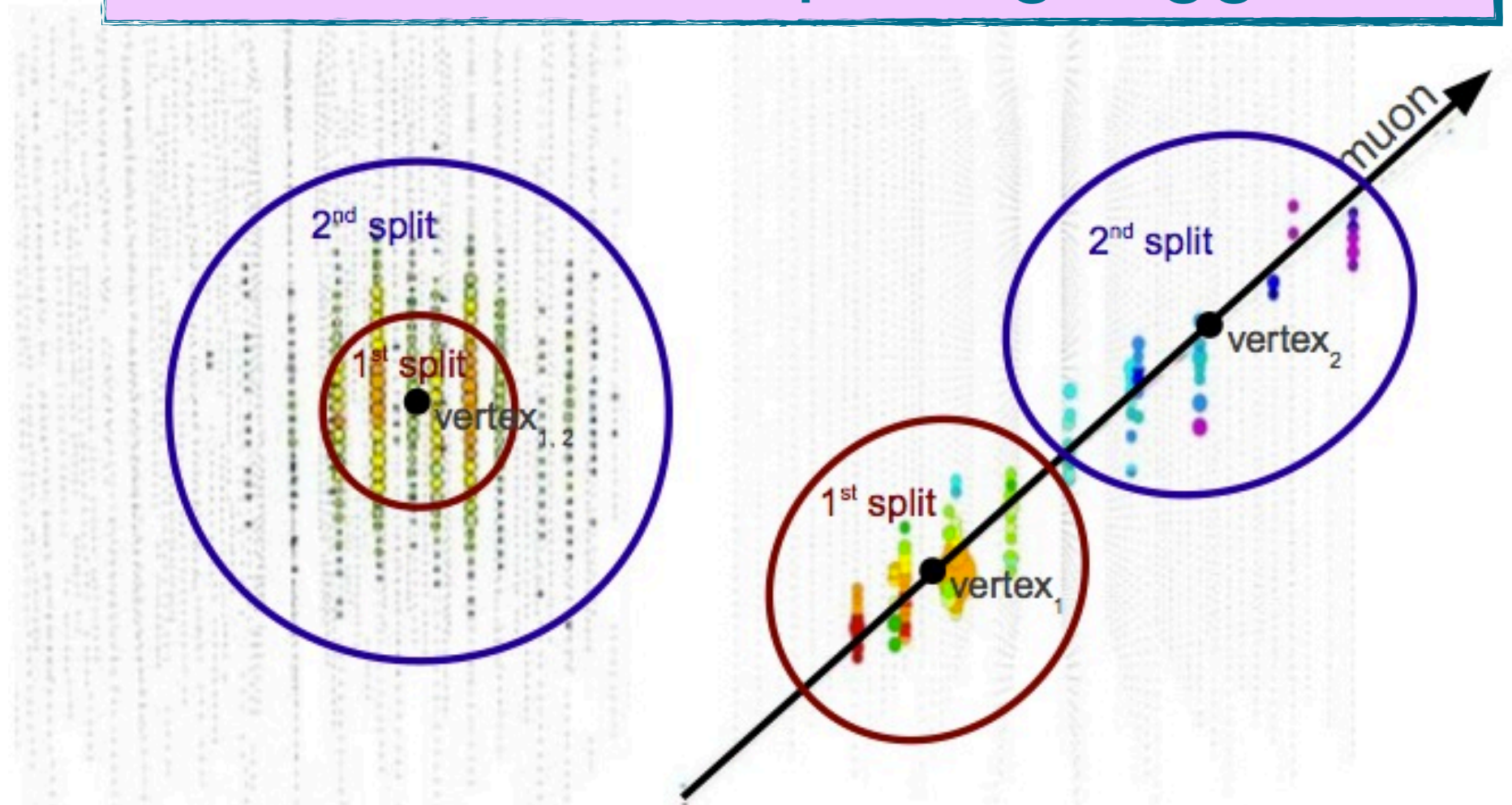
3. The trigger is set if the n. of satisfying hits is ≥ N'_{th}

Source declination:

★	$+90°$	⇔	North Celestial Pole
▲	$+47° \, 12'$	⇔	$90° - \lambda_{KM3}$
■	$+42° \, 48'$	⇔	λ_{KM3}
●	$+23° \, 26'$	⇔	Sun, summer solstice
✳	$0°$	⇔	Celestial Equator
○	$-23° \, 26'$	⇔	Sun, winter solstice
◇	$-29°$	⇔	Galactic Centre
□	$-42° \, 48'$	⇔	$-\lambda_{KM3}$
△	$-47° \, 12'$	⇔	$\lambda_{KM3} - 90°$
☆	$-90°$	⇔	South Celestial Pole

1. From **GPS** time of timeslice → **source direction**

2. L1 preselected events with **even one seed** are accepted.

3. **All hits** of each event are tested with the "**rotation**" procedure (**road-width** R_{max} restriction w.r.t. direction)

4. A *cluster* is formed when found N_{min} **consecutive hits**, **L1 seed included**.

5. If **time-overlap** among clusters → **clusters are merged** into one only bigger cluster.

6. **Small clusters** are treated with a **quick reconstruction** (to avoid accidental clustering of bkg)

7. The trigger is set if PMT *surface density* (w.r.t. the convex hull ⊥ direction) ≥ σ_{th}

1. Subdividing all the event hits in 2 time splitted groups
2. Vertexes reconstruction and vertex position discrimination

3. "Inertia" tensor eigenvalues ratio

$$I^{k,l} = \sum_{i=1}^{N} A_i(\delta^{k,l}\mathbf{r_i^2} - r_i^k r_i^l), \qquad \mathcal{T} = \frac{I_1}{I_1 + I_2 + I_3}.$$

Algorithms serving Physics

Motivation	T-Trigger	Simple Causality Trigger	Sky Scan Trigger	Tracking	Stack-Analysis	Vertex/ Inertia
muon	✔	✔	✔	✔	✔	✔
showers	✔					✔
slowly moving particles	✔	✔				✔
sources	✔			✔	✔	✔

L2 - **Follow-up trigger**

SuperAGILE, FERMI, SWIFT, INTEGRAL, MAXI/GSC, Suzaku, KonusWind, HETE 2

GRB alert system within the **GRB Coordinate Network**

> 1 GRB / 2 days

http://grb.sonoma.edu/

redshift ~1 · · · GRB

γ ν

t = -250 s

t = 0 s

satellite

t = 20 s

GCN

alert message

GRB trigger

requested at least 20s of buffered data
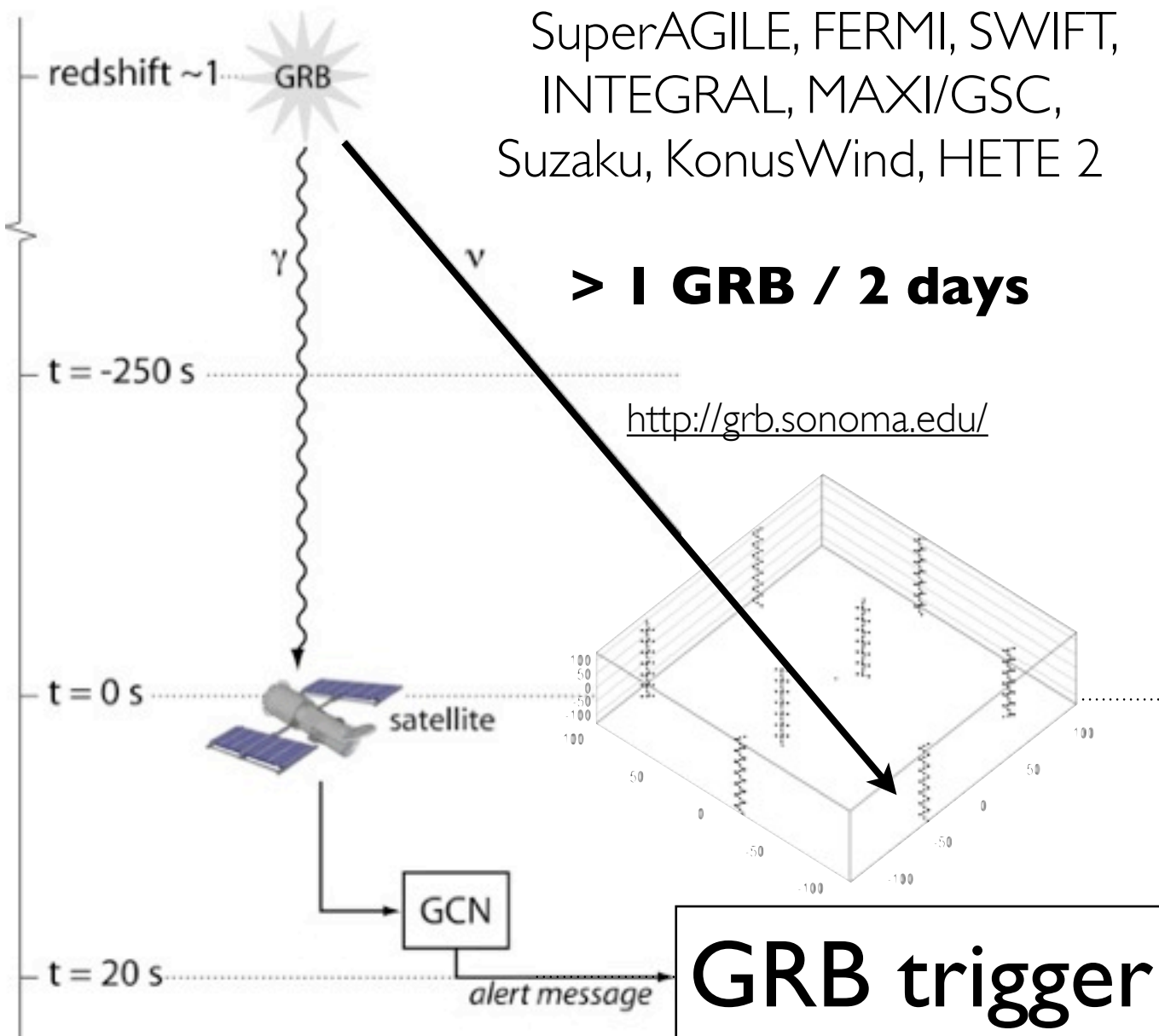
... but who is supposed to be buffering ?

... and who is supposed to apply a stack-analysis through the expected duration of a burst?