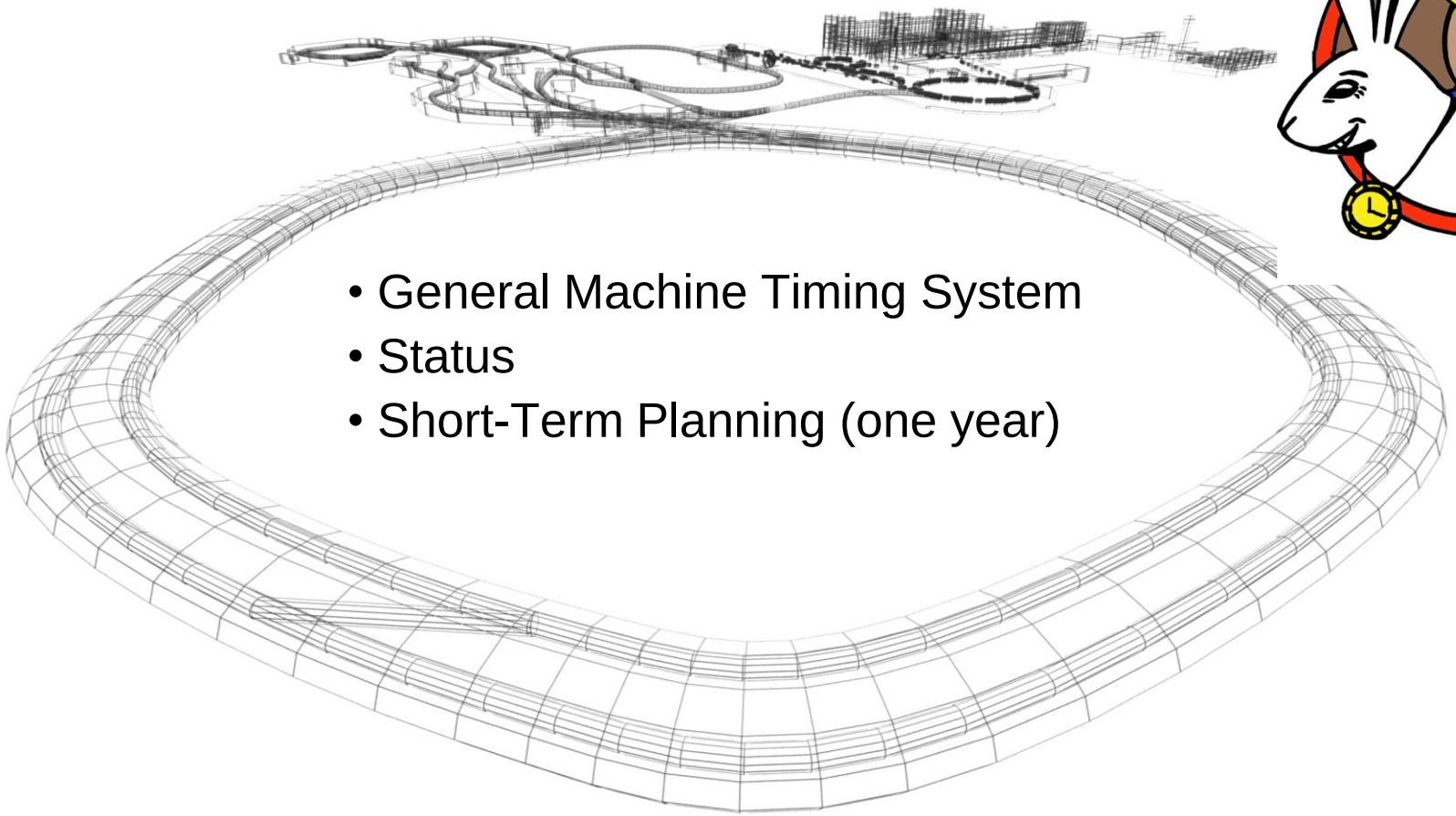
Status General Machine Timing System



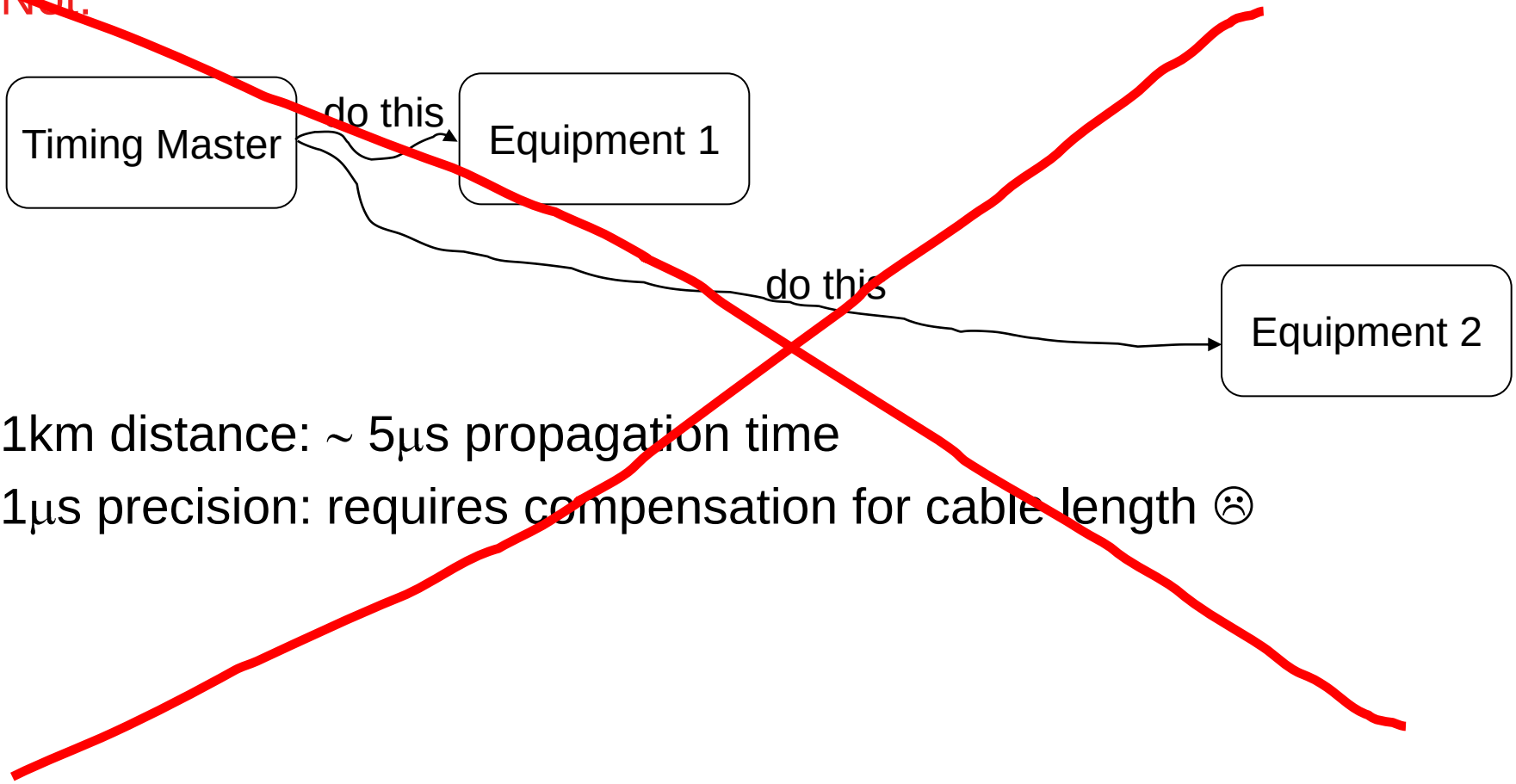
- General Machine Timing System
- Status
- Short-Term Planning (one year)

Acknowledgements

- GSI Timing Team: Marcus Zweig, Stefan Rauch, Mathias Kreider, Cesar Prados, Wesley Terpstra, Ralph Bär, Dietrich Beck
- CERN Timing Team: Tomasz Włostowski, Javier Serrano, Maciej Lipinski, Evangelia Gousiou, Erik van der Bij, Jean-Claude Bau, Pablo Alvarez, Greg Daniluk, …
- GSI/EE: Jan Hoffmann, Nikolaus Kurz, Holger Brand, …
- ...

Idea: Timing System Based on Time

Not:

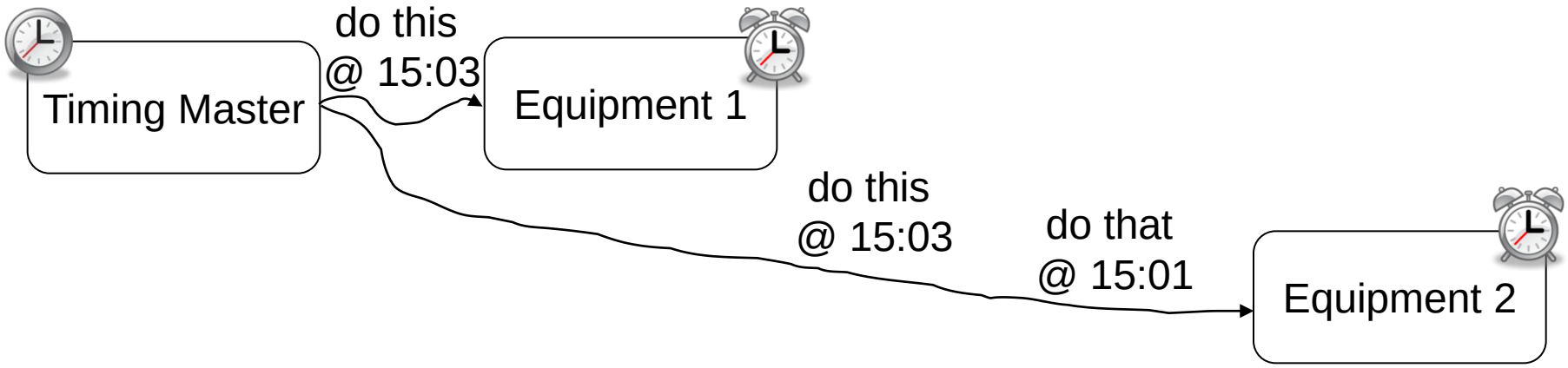


1km distance: ~ 5μs propagation time

1μs precision: requires compensation for cable length ☹

Idea: Timing System Based on Time

Instead:



- equipment pre-programmed for autonomous action at a given time
- action scheduled via timing-messages
- required: clock synchronization ~ns

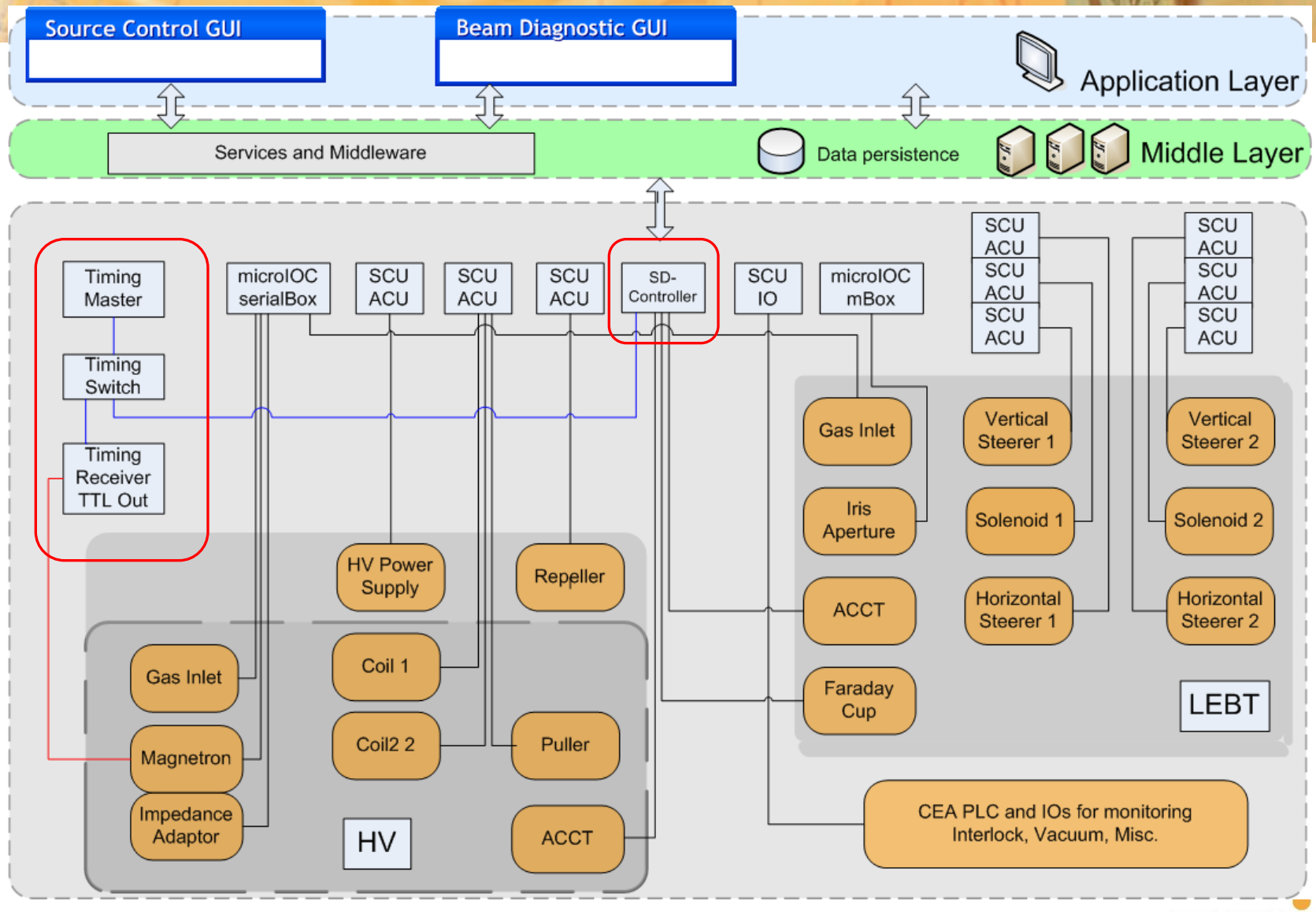
⇒ distribution of information and execution of action are decoupled !!!

- timing-events must be sent “early enough”
  - upper bound latency for transmission (e.g. 100 μs): real-time!
  - lossless transmission: robustness!

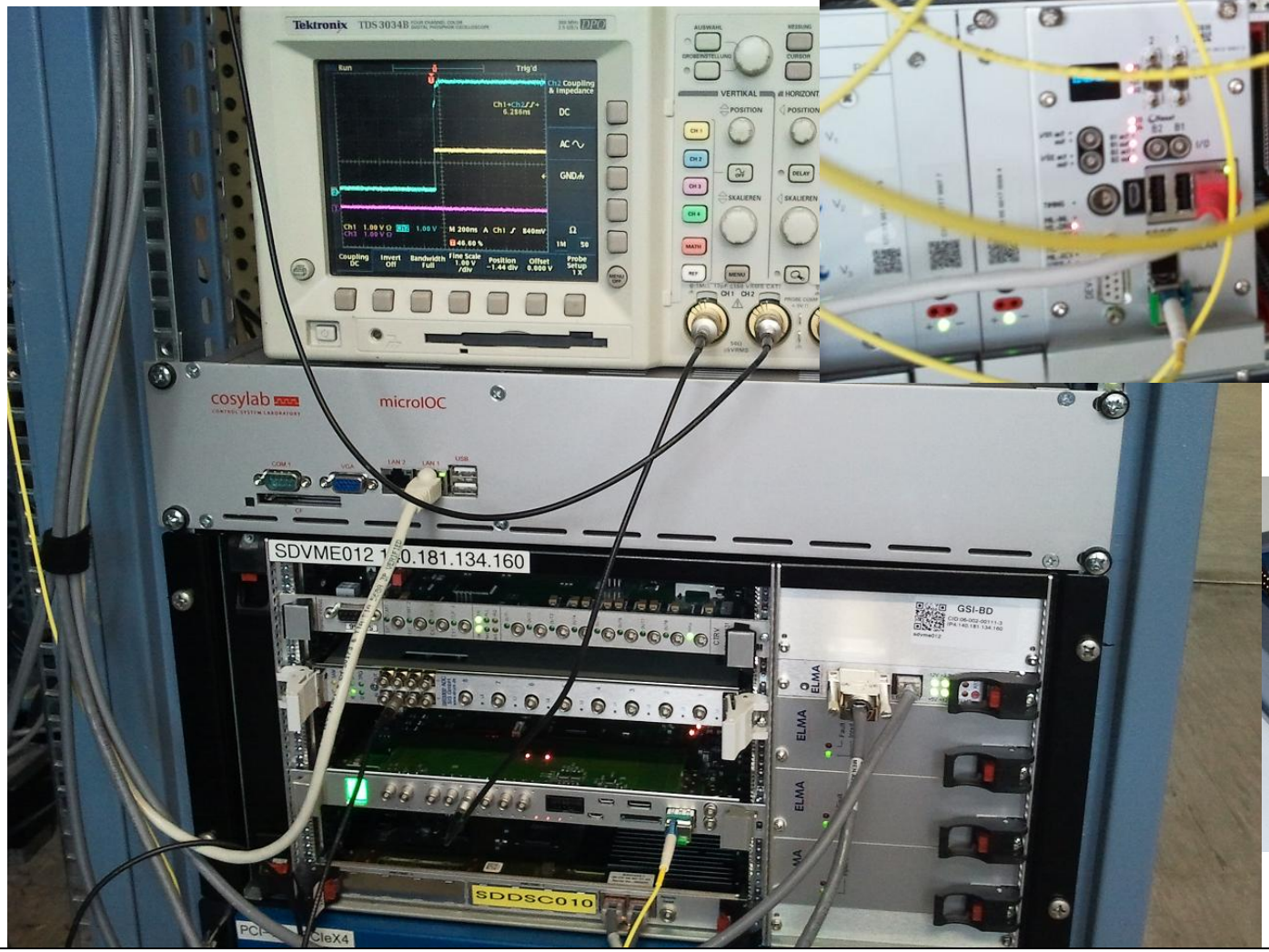
White Rabbit "Fieldbus" Cooking Recipe

- network: Gigabit-Ethernet
- PTP (Precision Time Protocol) IEEE1588-2008
  - free-running oscillators on network nodes
  - need to re-sync often: lot's of traffic, bad for determinism
  - only 1-100 μs synchronization of clocks
- SyncE (Synchronous Ethernet)
  - receiver's clock recovered from 125MHz carrier
  - 8ns precision
- precise phase measurement and adjustment
- (dedicated switches for clock propagation across a network)
- clock synchronization with sub-ns precision and low-ps jitter using optical fiber (no copper!)
- Philosophy: Open Hardware and Software (www.ohwr.org)

Milestone R1: Control System pLinac Source



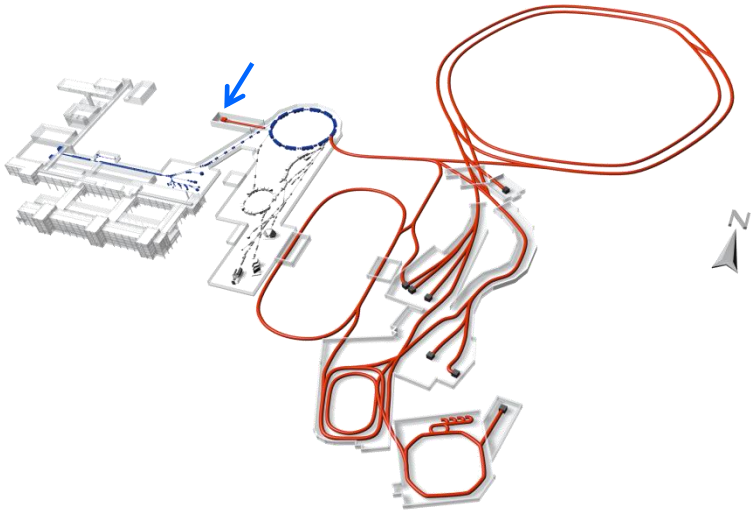
“A Very Sophisticated 4 Hz Pulser”

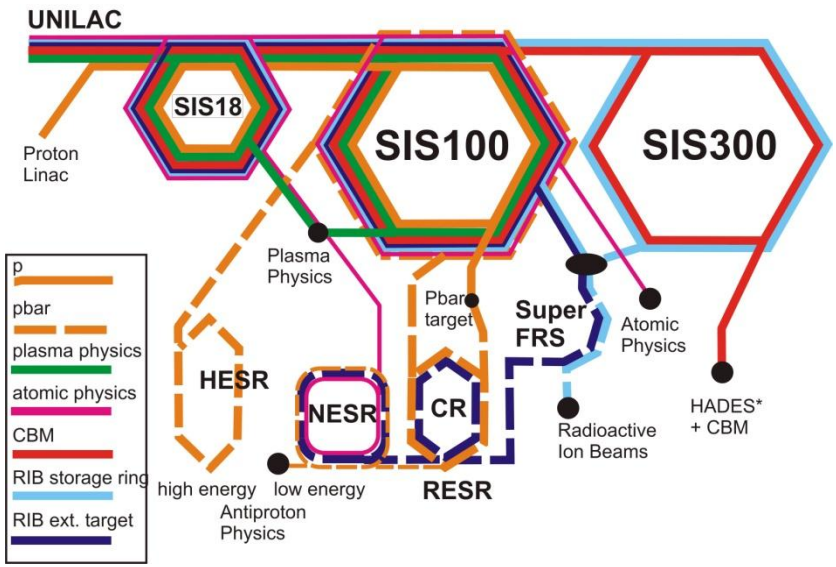


Status -“Proof of Principle”

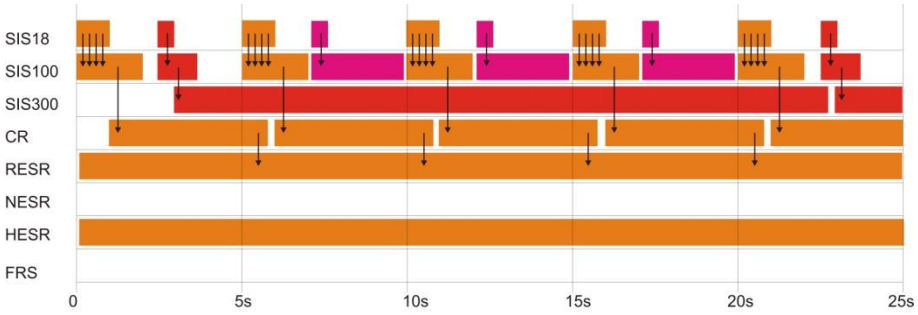
- Clock Master: using GPSDO and/or BuTiS as reference clocks
- Data Master (message generation):
  - one machine only, no interlocks et al.
  - using Scalable Control Unit (FAIR standard controller)
  - implemented in lm32 in FPGA (no OS, hard real-time)
- Network:
  - 18 port White Rabbit switches (production quality, from “7 Solutions”)
  - no Forward Error Correction, no redundancy, ...
- Nodes based on Altera FPGAs (signal/IRQ generation):
  - SCU: in-house development by CSCO, o.k.
  - EXPLODER2C: standalone, in-house development by CSEE, o.k.
  - VETAR1: VME, in-house development by CSEE, needs revision
- VHDL and Software
  - some key components exist, ...

General Machine Timing System (GMT) @ FAIR





- parallel execution of beam production chains
- cycles: 20ms to hours
- trigger and sync. equipment actions
- 1 µs precision in 99% of all cases
- few ns precision for kickers
- (few ps for rf-systems: BuTiS)
- many rings
- > 2000 devices connected to timing system
- large distances
- robustness: lose at most one message per year



Operation Mode #5: pbar in HESR, CBM in SIS300 and high energy Atomic Physics.

Implementation Plan for GMT

- iterative development
- release milestones twice a year
- closely linked to FAIR project management
- challenging schedule!
- soon: control system for CRYRING (“R2”, “R3”)
- long term: obtaining hardware
  - control system nodes: ~1200-1500 SCUs, GSI/CSCO
  - network: ~ 300 switches, commercial product (OWHL)
  - nodes for beam-diagnosis: ~500, in-kind contribution
  - special purpose hardware, GSI/CSEE
- GSI Timing Team
  - Integration
  - VHDL and Software

What about DAQ?

According to the specs, the GMT shall support DAQ

- presently VME as only form factor
  - not yet mature, board requires revision
  - drivers for different types of CPUs (RIO, MEN020) in progress
- globally triggered DAQ systems: time stamp latching
  - VHDL for Time Stamp Latch Unit developed by Mathias Kreider, GSI
  - integration (I/O, API,...) missing
- triggerless DAQ: time stamp fan-out, BuTiS pulse identification
  - VHDL code developed by Peter Schakel, KVI
  - integration (I/O, API,...) missing

Present Situation for DAQ

- components almost exist
- application in 2014 beam times desirable, but
- integration work is required
- wild guess: ~ 6 months full time at GSI
- controls department (CSCO): higher priorities on other components that are closely linked to the next release milestones
- electronics department (CSEE): higher priorities on other projects

➔ time stamp latching and distribution won't be done in time, except somebody volunteers to help

Thanks for your attention...

BTW: 5 year position within the GSI Timing Team (gsi.de or stepstone.de)

Timing Network

- Determinism
  - “timing-events shall arrive early enough”
  - upper bound latency from timing master to nodes
  - no handshake: use UDP or raw Ethernet (no TCP)
  - switches implement cut-through for high-priority messages
- Robustness – Forward Error Correction
  - bit errors: Hamming code
  - packet loss: Reed-Solomon code
- Redundancy
  - redundant links between switches
  - link aggregation (in combination with Reed-Solomon)

➢ **Dedicated network for timing only**

➢ **Dedicated infrastructure (switches, …)**

Timing Systems @ FAIR

General Machine Timing System
- based on White Rabbit
- fairly cheap: € 800,- SPEC board
- sub-ns synchronization
- distribution of
  - (clock)
  - time-stamps
  - timing-events
- BEL group

- one global timing master
  - clock derived from BuTiS

Talk T. Fleck, FAIR Technikforum

Bunch phase Timing System (BuTiS)

- invest € 15k per receiver station
- ps precision (100ps/km accuracy)
- low phase noise,
- distribution of
  - clocks!!!
  - NO time-stamps at FAIR
  - NO timing-events
- HF group

- one global BuTiS center

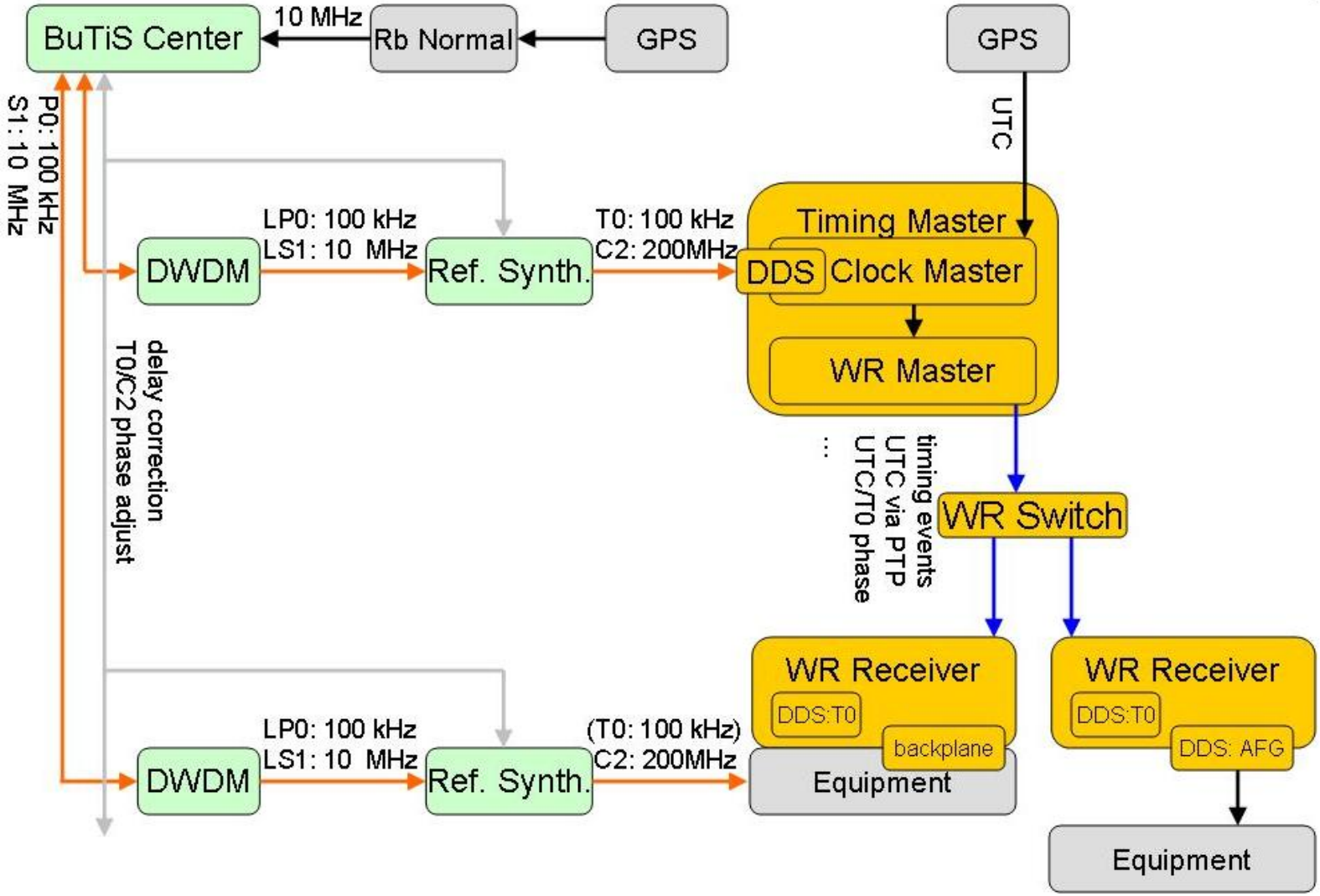
Talk P. Moritz, FAIR Technikforum

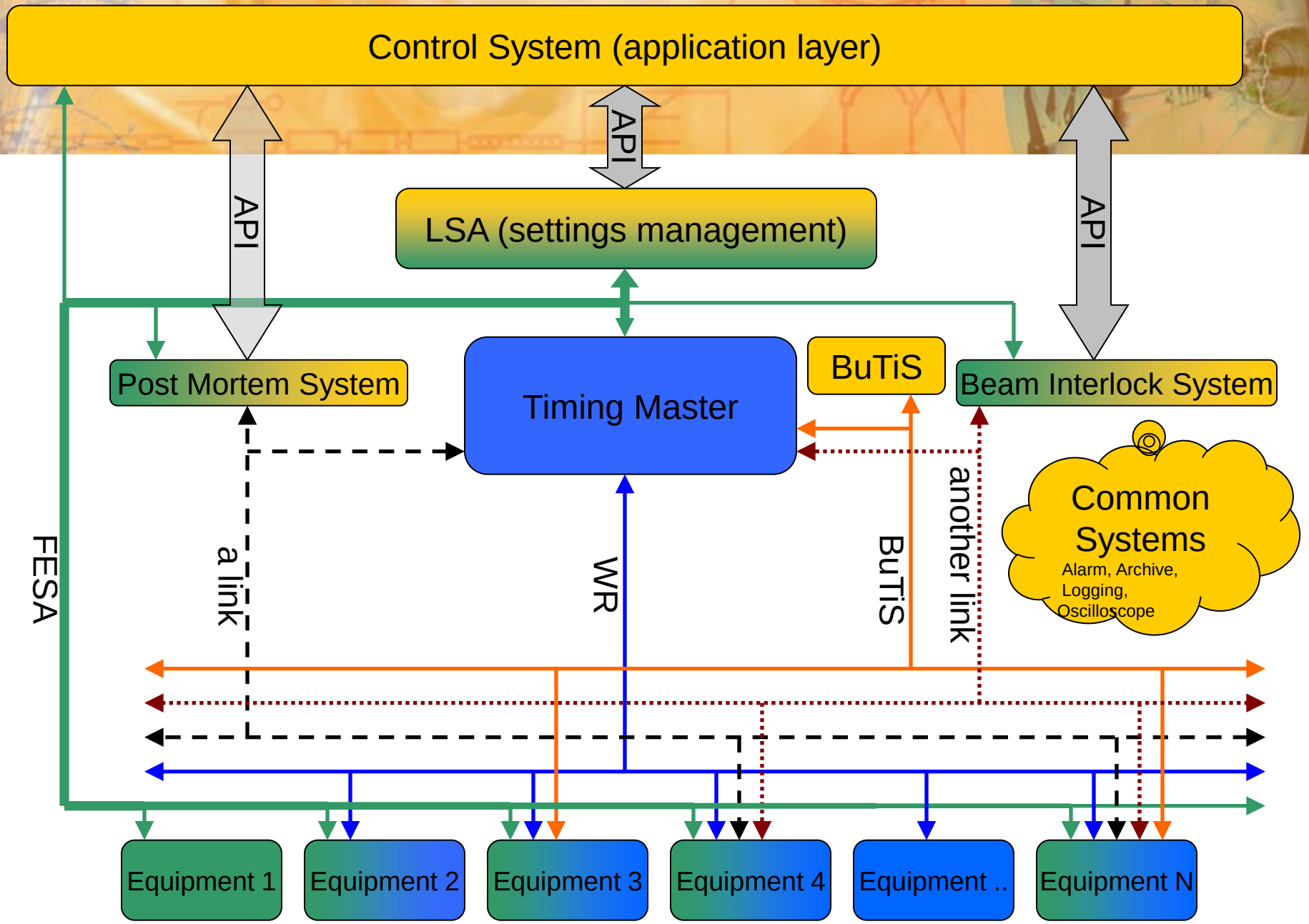
Remark: If you are interested time-stamps, you need White Rabbit.

If you are interested in the most accurate clocks, you need BuTiS.

If you are interested in both, you need both.

BuTiS and GMT





Disclaimer: This is my personal view and may be incorrect and not up-to-date

Summary GMT…

- no general purpose network
- physically separated from other networks (gateways)
- distribution of clock, time and events from timing-master to nodes
- campus-wide distribution of information - not restricted to the accelerator control system
- only the timing-master may send data to the network[1]
  - exceptions for specific use-cases of the accelerator control system
  - tentatively: user-nodes may send data within VLANs

[1]Except WR-PTP and replies to read-requests from the timing master

F-DS-C-06e: Detailed Specification of the FAIR Accelerator Control System Component „Timing Receivers“

- receive (and react to) “Events”
  - Timing-Events
  - Info-Events
  - Commit-Events, Post-Mortem,…

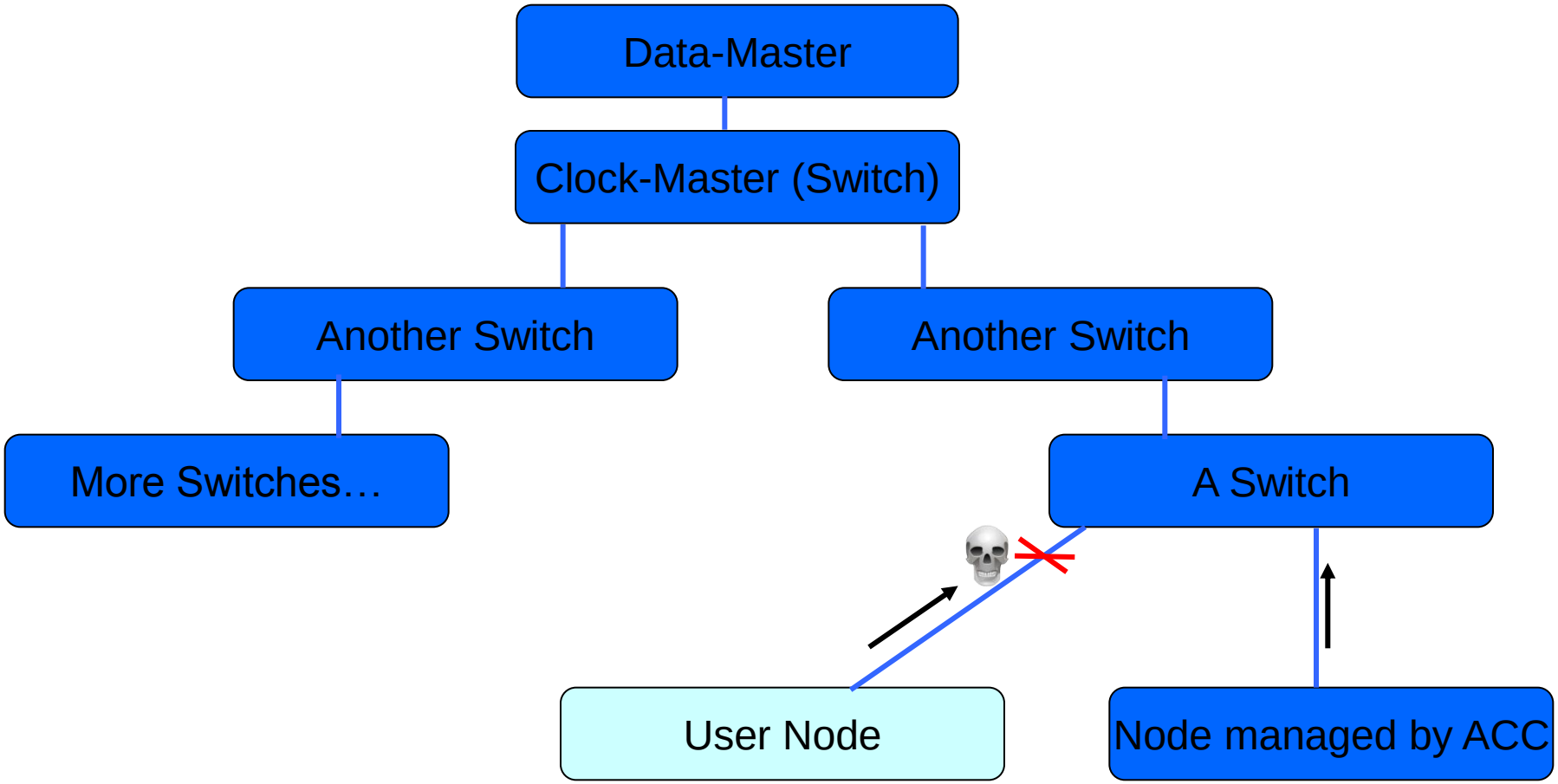
- TR_190: Timestamp latching (globally triggered DAQ),
  tentative: precision 3ns or better, requirement for unique assignment of c2 clock edges (BuTiS)

- TR_200: BuTiS clock re-generation (T0’, c2’)
  low precision: 3ns or better

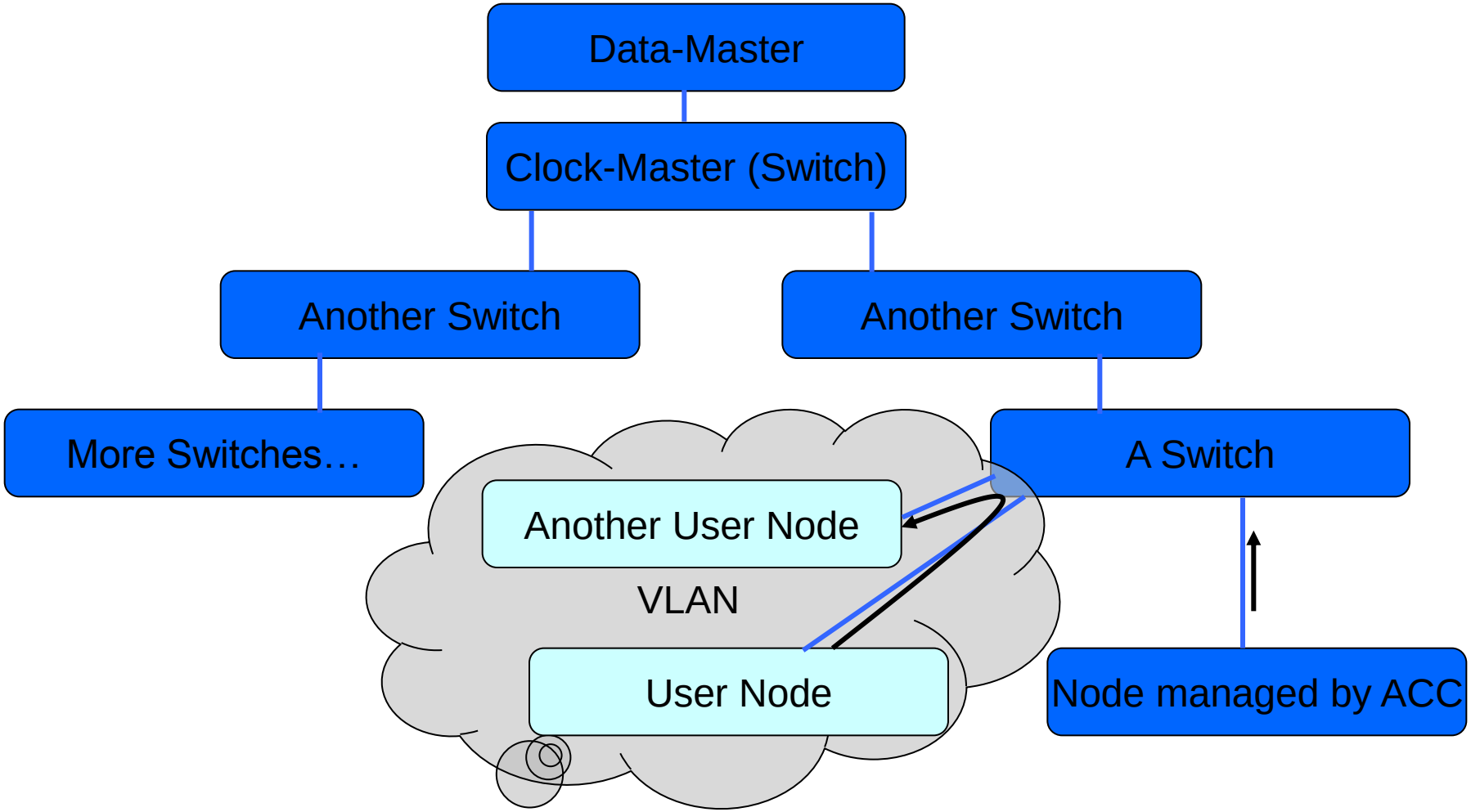
- TR_210: Timestamp distribution (triggerless DAQ), T0’ clock ticks + “at the next tone it will be …”

F-DS-C-05e: General Machine Timing System, tentative user interface - no commitment

Network


Data-Master
Clock-Master (Switch)
Another Switch
Another Switch
More Switches…
A Switch
User Node
Node managed by ACC

Network


Data-Master
Clock-Master (Switch)
Another Switch
Another Switch
More Switches…
A Switch
Another User Node
VLAN
User Node
Node managed by ACC

Ports and Switches

- the number of ports is requested via the building coordinators
- collecting requests and the planning is already done
- a few ports will be provided in experimental areas by the GMT free of charge
- number of available ports depends on the local situation
- in case more ports are required
  - additional switches provided and maintained by ACC
  - costs shall be covered by user
  - adding additional layers of switches may impact determinism

“Anfordereinheiten” - (Local Beam Request Units)

- to be discussed
- new concept required for FAIR
- as before: dedicated box
- connected to the timing network
- managed by ACC
- display
- I/O

Summary User Interface GMT

- timestamp distribution and/or latching
- receive all events distributed by the timing master
- user data must not be sent via the timing network
- user data may be sent within local VLANs (if available)
- local beam request units should be provided by ACC

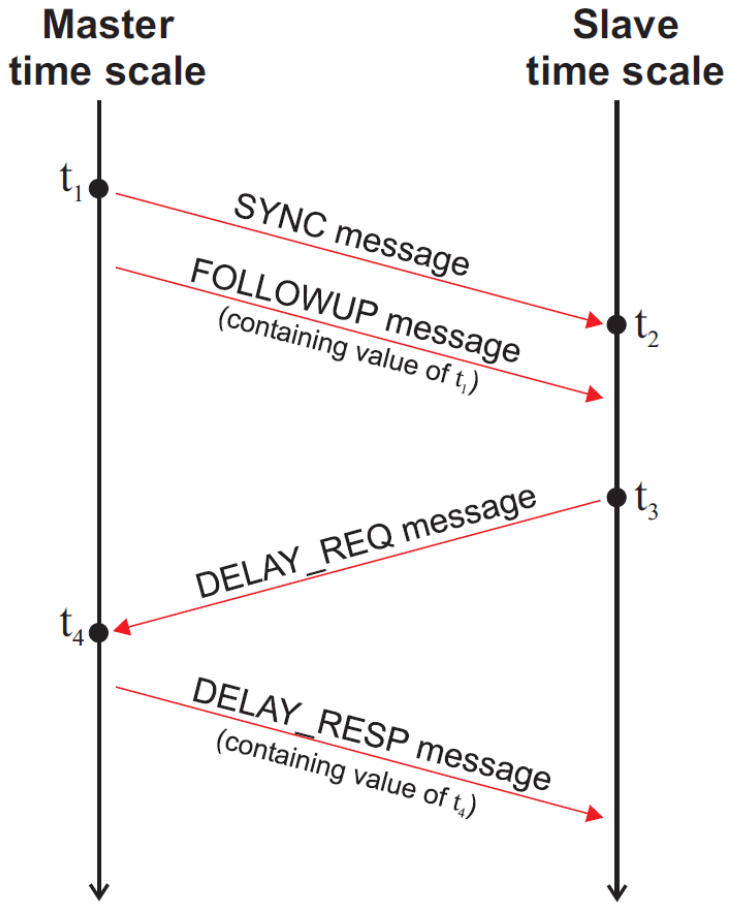
view towards SIS 100, 23 November 2011

Further reading

- www.ohwr.org
- https://www-acc.gsi.de/wiki/Timing/TimingSystemDocumentation
  (search for “FAIR timing system”)

White Rabbit Cooking Recipe I: Clocks

- network: Gigabit-Ethernet
- PTP (Precision Time Protocol) IEEE1588-2008



with value t_1.. t_4 one can

- **calculate one-way link delay**

$$\delta_{ms} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$$

- syntonize clock rate by tracking the value

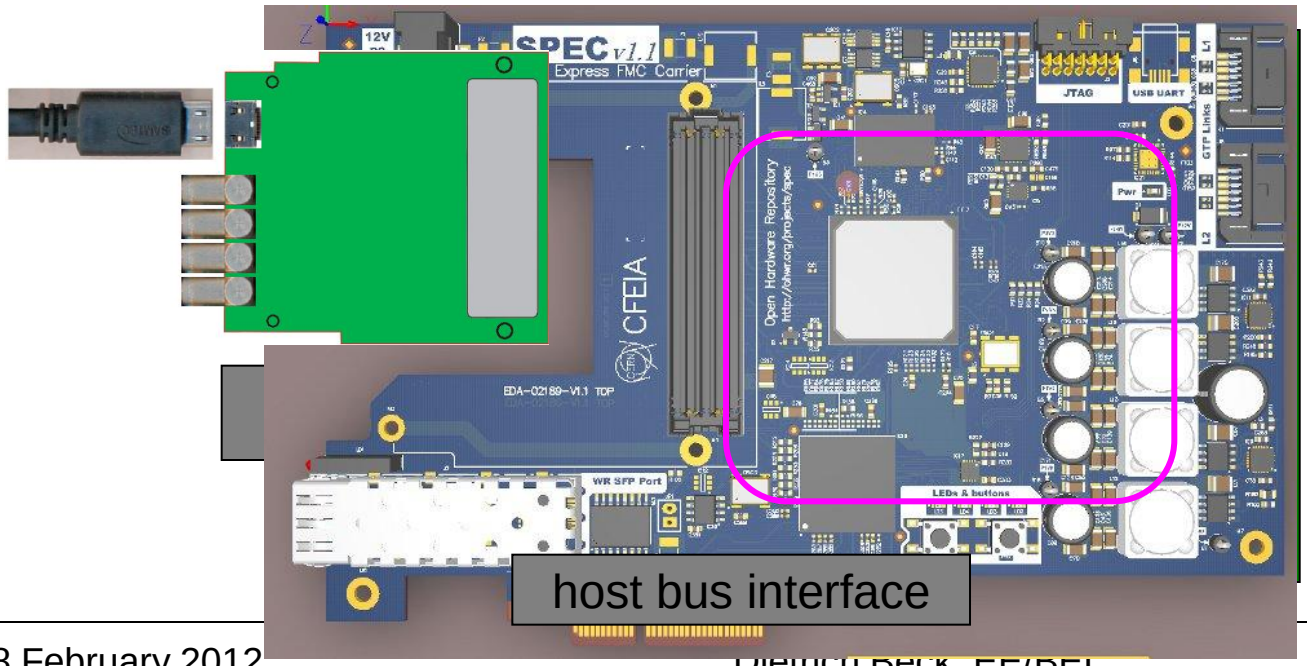
$$t_2 - t_1$$

- **calculate clock offset**

$$offset = t_2 - (t_1 + \delta_{ms})$$

WR Cooking Recipe III: Implementation…

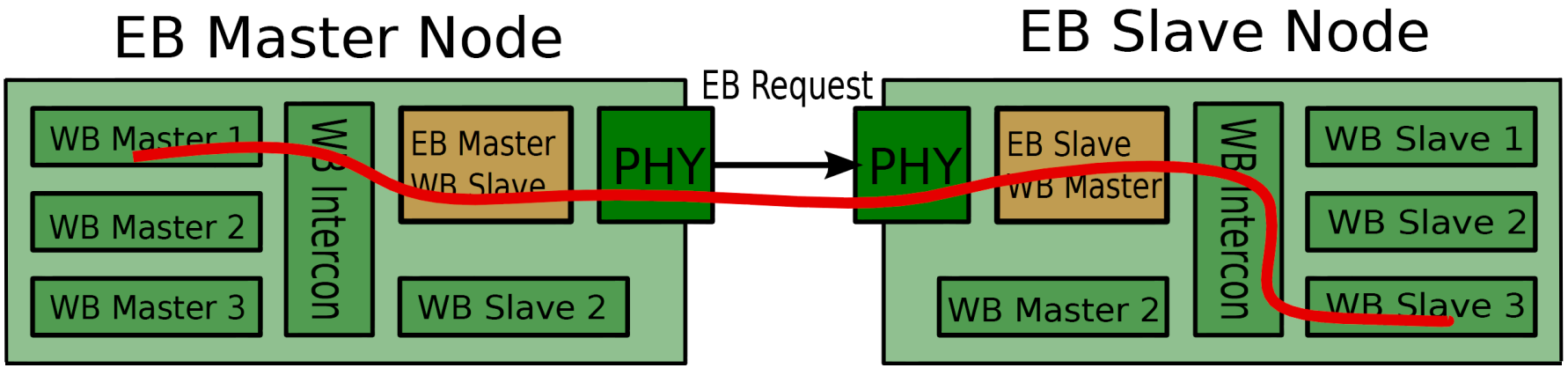
- path asymmetry (two cables) impacts phase compensation
- ➢ White Rabbit PTP
  - one bidirectional single-mode optical fiber
  - different wave lengths for TX/RX: speed differs slightly (dispersion)
  - link-delay-model
    - relative delay coefficient α depends only on the type of fiber (off-line)
    - fixed delays are measured on-line
- White Rabbit node



Inside the FPGA:

Transmission of Timing-Events: Etherbone

- idea: Connect distant Wishbone buses via a network protocol layer. Ethernet + Wishbone = Etherbone
- direct memory access to devices via White Rabbit network
- used for timing-events



…Next Steps

@GSI/CERN: WR starter kit (timing system demo) spring 2012

@Saclay: Low energy part of p-linac: first WR based timing system provided by GSI (early 2013)

@CERN:

- Installation of WR based timing system at AD ring (2013)
- Installation of WR link between CERN and Gran Sasso (2012)

Status

- Slovenia: In-Kind contribution timing receivers. Receivers and FPGA building blocks for FAIR timing system specified.
- Germany/GSI: In-Kind contribution General Machine Timing System: specification until March 2012.

- White Rabbit PTP link between two nodes established
- sub-ns synchronization with a jitter of ~hundred ps
- propagation of WR PTP over three layers of switches
- FPGA building blocks for simple timing system exists
- integration work in progress
- planning of WR network cabling in progress

- First WR nodes developed by EE