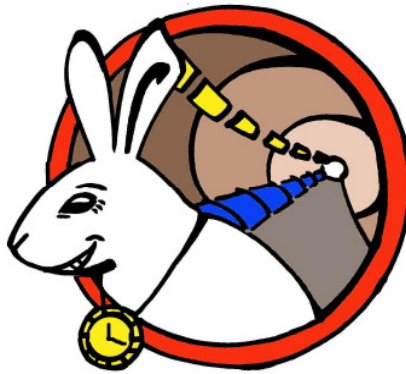


# The General Machine Timing System for FAIR and GSI

Dietrich Beck, Mathias Kreider, Cesar Prados, Wesley Terpstra, Stefan Rauch, Marcus Zweig

## Abstract

This document summarizes the "General Machine Timing System", which is a component of the accelerator control system.



## Table of Contents

1	Purpose and Classification of the Document .....	3
1.1	Classifications of Requirements.....	3
2	Scope of the Technical System .....	3
2.1	System Overview.....	3
2.2	Limits of the System and Environment.....	5
2.2.1	Limits .....	5
2.2.2	Interfaces.....	5
2.2.3	Environment .....	7
2.3	Basis of Concept.....	8
2.3.1	Functional Requirements.....	8
2.3.2	Non-functional Requirements .....	12
2.3.3	General Constraints .....	13
2.3.4	Architectural Principles.....	13
3	Technical Specifications .....	13
3.1	The General Machine Timing System .....	13
3.2	GMT Components .....	15
3.2.1	Timing Master .....	15
3.2.2	Timing Network.....	16
3.2.3	Timing Receiver Nodes.....	17
3.2.4	MAC and IP Addresses.....	17
3.2.5	Announce Messages and Local Delay Compensation .....	17
3.2.6	GMT Supporting Systems .....	18
4	Acronyms and Definitions .....	19
I.	Related Documentation.....	20
II.	Document Information.....	20

## List of Tables

Table 1: List of Functional Requirements .....	12
Table 2: List of Non-functional Requirements.....	12

## List of Figures

Figure 1: Architecture Overview of the Accelerator Control System.....	4
Figure 2: Overview - GMT and some of its interfaces (see text).....	6
Figure 3: Jitter, precision (black signal), and accuracy (green signal) in the scope of the GMT.....	8
Figure 4: Sketch of the GMT (see text).....	14
Figure 5: Usage of Soft-CPU's in the data master.....	15

## 1 Purpose and Classification of the Document

The purpose of this document is to describe the accelerator control system component "General Machine Timing System" [24] but without the work package "Timing Receivers".

### 1.1 Classifications of Requirements

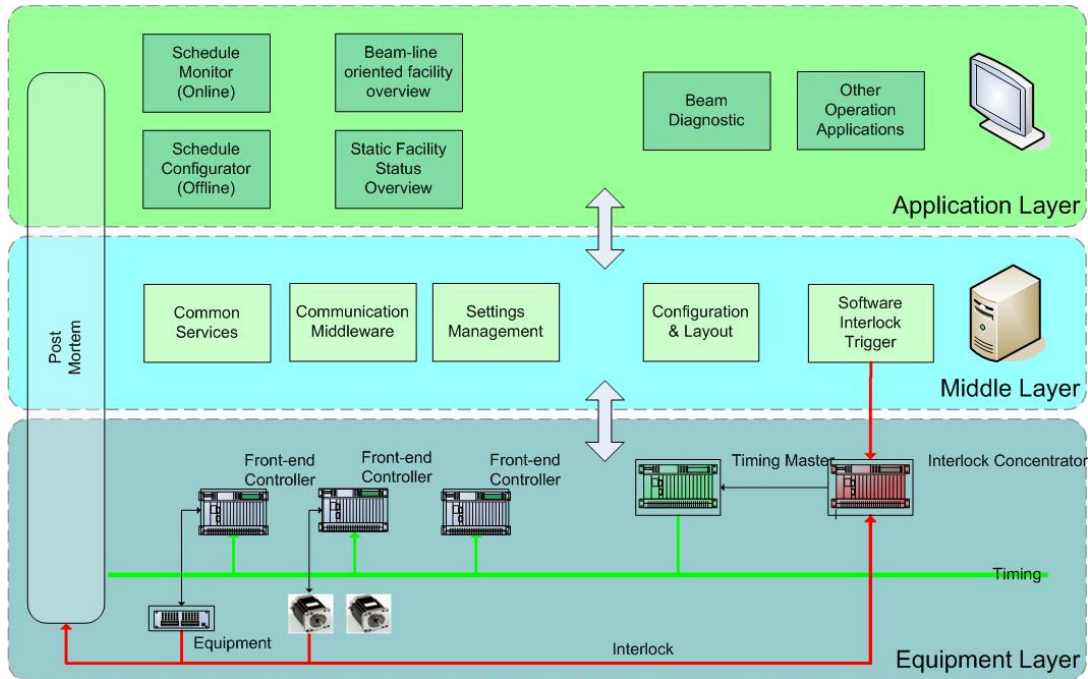
The following definitions of requirement classifications are being used throughout the document:

- "**Must**" or "**shall**" or "**is required to**" is used to indicate mandatory requirements, strictly to be followed in order to conform to the standard and from which no deviation is permitted.
- "**Must not**" or "**shall not**" mean that the definition is an absolute prohibition of the specification.
- "**Should**" or "**is recommended**" is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others or that a certain course of action is preferred but not required.
- "**Should not**" or "**is not recommended**" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighted before implementing any behavior described with this label.
- "**May**", which is equivalent to "**is permitted**", is used to indicate a course of action permissible within the limits of the standard.

## 2 Scope of the Technical System

### 2.1 System Overview

The purpose of the General Machine Timing System (GMT) is to trigger and synchronize equipment and software actions, timed according to the accelerator cycles. The timing system must handle machine cycles from milliseconds up to the order of several seconds for synchrotrons and up to several hours for storage rings. Beam manipulation phases may be as short as a few micro seconds. The timing system also must take real time decisions to decide correctly between predefined alternative cycles to be executed [1].



**Figure 1:** Architecture Overview of the Accelerator Control System.

Indicated in Figure 1 are two sub-systems of the GMT in the overall context of the accelerator control system, a *Timing Master* and a *Timing Network*, which is dedicated to the GMT. As shown, the GMT is contained in the lowest of the three layers of the accelerator control system, the so-called *Equipment Layer*.

The timing master is a logical device and receives a schedule for the operation of the FAIR accelerator complex from the *Settings Management*. The timing master schedules actions of the *Equipment* connected to the *Front-end Controllers* (FEC) for timely execution by sending so-called *timing messages* through the timing network.

The *Application Layer* and the *Middle Layer* only request what the FAIR accelerator complex should do and transmit set values to the *Equipment Layer*. The actual beam production is controlled by the GMT, which, like the on-board computers on a space craft, autonomously operates the facility in real-time.

The task of the GMT within the accelerator control system is best described in the picture of an orchestra performing a piece of music. Notes are written by a composer (application layer and the middle layer of the control system). The sheets of music (set values) are distributed to the musicians (front-end controllers) and to the conductor (timing master). Finally, the conductor uses his baton (timing network) to orchestrate the efforts of the musicians.

The GMT triggers and synchronizes about 2000 connected FECs and other equipment covering a distance of up to 2 km between the nodes in the timing network. Although a precision of 1  $\mu$ s is sufficient in most cases, some equipment like kickers need timing with nanosecond precision. An important issue is the link between the GMT and the timing system of the existing facility.

At FAIR, the GMT is complemented by the Bunch phase Timing System (BuTiS), which is a high-precision clock distribution system mainly targeted at the radio-frequency systems of the accelerator [10]. By convention, the term “timing system” is used only for the GMT [1].

## **2.2 Limits of the System and Environment**

### **2.2.1 Limits**

The timing system relies on a dedicated timing network which is distinct from the general purpose communication network of the accelerator control system. The timing system is not intended for general purpose communication, transmission of bulk data and information not related to the timing system.

All nodes of the timing system are trusted and security aspects must be handled by the interfaces and gateways to other systems or networks.

Protection of machines and safety of personnel are handled by the Machine Protection Systems [4] and Personnel Safety System [5]. However, the GMT should assist the Interlock and Post Mortem systems for implementing some aspects. These are specific cases which are described later in this document.

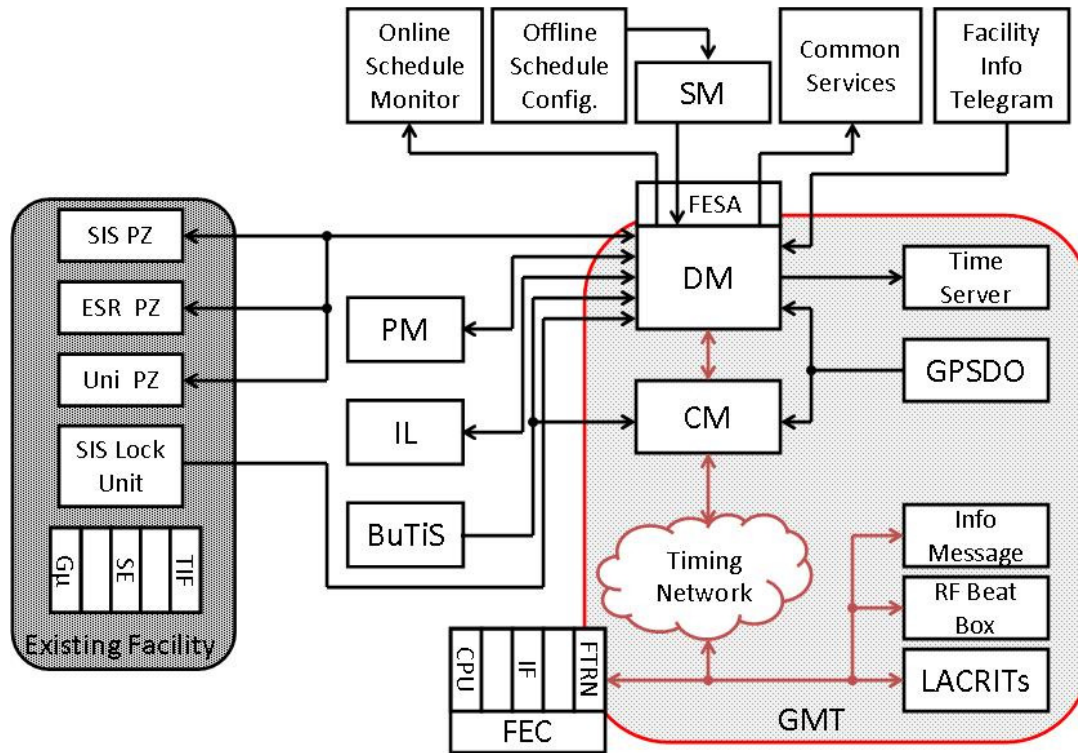
The timing master has no prior knowledge about machine schedules and alternatives. It just executes schedules that are provided by the Settings Management System [3].

The interface of a FEC towards the timing system is called FAIR Timing Receiver Node (FTRN).

- The hardware of the FTRNs is out of the scope of this document.
  - Most of the front-end systems at FAIR include a Scalable Control Unit (SCU). The SCU is provided by the component Equipment Interface and Control [6].
  - Other form factors than the SCU including PCIe or VME are provided by the component Timing Receivers [7].
- Some parts of the software and HDL of the FTRNs are out of scope of this document. The out-of-scope parts are described in the detailed specification Timing Receivers [7]. Remark: Other parts of software and HDL and, most importantly, its integration to an FTRN and the GMT are in-scope.

### **2.2.2 Interfaces**

The GMT has several interfaces to other subsystems of the accelerator control system.



**Figure 2:** Overview - GMT and some of its interfaces (see text).

Figure 2 shows the GMT and its main interfaces. The GMT includes a *Data Master* (DM) and a *Clock Master* (CM), a dedicated *Timing Network* and integrates FTRNs. Furthermore, the GMT includes GMT Supporting Systems (GMTS), namely a *Time Server*, a *GPS Disciplined Oscillator* (GPSDO) as time normal, an *Info Message Gateway* and *Local Accelerator Control Request Interfaces* (LACRITs). The main parts of the GMT are surrounded by a red box. Interfaces between the timing system and other systems are described in the following.

**Settings Management System (SM).** The SM is an off-line system and has the task of generating consistent data for the synchronized operation of all devices and machines. The SM communicates planned execution schedules and changes to these schedules to the GMT. The GMT shall provide an interface which the SM must use [3].

**Post Mortem System (PM).** The PM is a diagnosis tool for analyzing the origin of problems leading to equipment failure and consequent loss of beam. In such an incident, the PM provides a complete snapshot of the machine for further analysis, including the time span before and after an incident. A snapshot requires freezing, unfreezing and collecting data in the FECs. Freeze and unfreeze actions in the FEC are triggered by *post-mortem-events*. The GMT must provide an interface to the PM for distributing the trigger to connected FECs [8].

**Interlock System (IL).** The IL concentrates various interlock source signals, processes them and triggers the appropriate actions [9]. IL and GMT are interfaced. In case of an interlock signal, the GMT changes the schedules of the machines according to contingency plans provided by the SM. The GMT must provide an appropriate interface to the IL. In addition, the GMT itself is a source to the interlock system.

**Bunch Phase Timing System (BuTiS).** BuTiS provides high-precision radio-frequency (RF) synchronization by distributing a 100 kHz ident clock (T0), a 10 MHz reference clock and a 200 MHz sine clock (c2) [10]. The clock and time stamps of the GMT will be derived from BuTiS clocks. The GMT must provide an interface to other systems for re-synthesizing and distributing clocks derived from the T0 and c2 clocks.

**Existing Facility.** The control system modifications and other control system means which are needed to integrate the existing facility into the FAIR environment are described in [11]. The GMT must provide an interface to the existing facility and integrate or replace parts of the control system of the existing facility.

**Front-End Controllers (FEC).** At the lowest level of the control system, equipment interfaces must be accessed by FECs to communicate with the equipment [1]. In order to synchronize actions on front-ends, the GMT must provide an interface to the FECs. The interface are FTRNs and an API.

**Local Accelerator Control Request Interfaces (LACRITs).** Interface equipment to the accelerator control system must be available in local experiment areas. As an example, LACRITs serve for requesting beam from experiment areas or provide trigger signals for control or data acquisition systems. The GMT must implement LACRITs which are part of the GMTS systems.

**RF Beat Box.** This is some equipment that has the task of determining the phases of the ion bunch, bucket and RF-phases, which is required for transferring an ion bunch from one ring machine into an empty bucket of another ring machine. Typically, this involves a method using the beat frequency of slightly detuned frequencies in both rings. The GMT must provide an interface to the equipment involved.

**Info Message Gateway.** It must be possible to send the information distributed by the GMT to other networks via a gateway. By this, nodes not connected to the timing network may receive the information too. The Info Message Gateway is part of the GMTS systems.

**GPS Disciplined Oscillator (GPSDO).** A GPSDO system is mainly intended as a backup, when the interface to BuTiS is not required or not available. The GPSDO is part of the GMTS systems.

**Time Server.** The timestamps distributed via the GMT may have different values than the one provided by the time servers of the general infrastructure. The GMT must provide a dedicated time server that distributes the timestamps used by the GMT. The time server may implement NTP and is part of the GMTS systems.

**Common Services.** The GMT must provide an interface to the alarm system, the archiving service, diagnostic logging, operational logging and the software oscilloscope system [1]. When equipment of the timing system is represented by a FESA class, the interface to the Common Services shall be provided by FESA properties of the timing system equipment. The GMT also assists the Common Services by making timestamps available through the FTRNs.

**Facility Info Telegram Service.** The GMT must provide an interface which can be used for distributing important information via the GMT. An example of such information is the beam intensity measured upstream. Such information can be used to protect equipment of the beam diagnostics or for adjusting settings of the machine when space charge starts to play a role.

**Online Schedule Monitor (OSM).** This user application has the task of providing an online view of the GMT activities [12]. The GMT must provide an interface to this application.

### 2.2.3 Environment

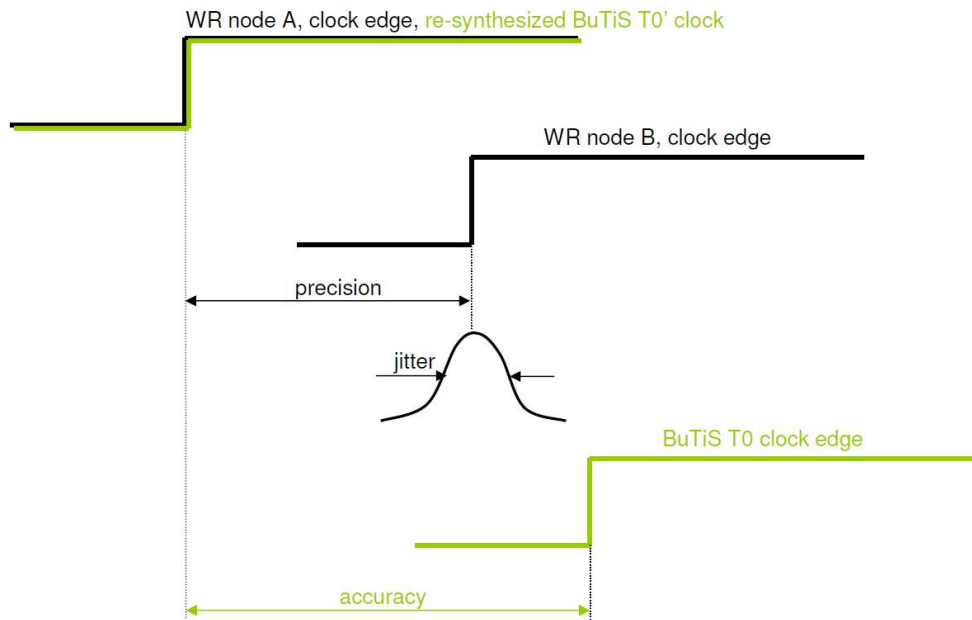
The GMT will be implemented using White Rabbit [13][14] which is an extension of Ethernet based on PTP and specific hardware for clock and phase synchronization. Hardware components, like the timing master shall be installed in electronic rooms in the facility. Electronic components have to run in the general environmental conditions of the FAIR maintenance buildings. Components of the timing network shall be installed in the environmental conditions that are equivalent to the ones for the general purpose network.

Components are not radiation hardened and should not be installed in the tunnels, caves or locations with significant radiation levels. If so, reliable operation is not guaranteed.

### 2.3 Basis of Concept

The GMT is based on the White Rabbit standard and adds features including determinism and robustness. Clocks and timestamps are propagated from a common grandmaster clock to all nodes in a dedicated timing network. A central master is used for transmission of time critical data with high priority through the network to the nodes. The nodes timely execute scheduled actions autonomously.

This document uses the terms *jitter*, *precision*, *accuracy*, *granularity* and *time*. The meaning of the first three terms is shown in Figure 3.



**Figure 3:** Jitter, precision (black signal), and accuracy (green signal) in the scope of the GMT.

**Precision** is the deviation of clock edges, when comparing a clock between any two nodes in the timing network.

**Jitter** is the short term fluctuation of the clock edge on a WR node around its mean value.

**Accuracy** is the deviation between the edges of a BuTiS T0 clock and the re-synthesized signal T0' on any WR node in the timing network.

**Granularity** is determined by the FPGA clock cycle of a WR node. A clock cycle has the length of 8 ns.

**Time.** The time standard of the GMT is defined by BuTiS clocks. One second has the length of 100,000 T0 clock cycles. Note: The BuTiS clocks are synchronized to a primary reference time source such as a GPSDO.

#### 2.3.1 Functional Requirements

The GMT fulfills the following functional requirements:

Number	Description of the Requirement
TS_010	<b>Equipment Triggering and Synchronization</b> – The GMT shall trigger and synchronize equipment actions.



TS_020	<b>Parallel Execution</b> – The GMT shall support parallel execution of many beam production chains.
TS_030	<b>Varying Cycle Times</b> – The GMT shall support machine cycles from 20 ms to many hours.
TS_040	<b>Varying Manipulation Phases</b> – The GMT shall support machine manipulation phases from microseconds to many hours.
TS_050	<b>Scalability</b> – The GMT must provide scalability appropriate for 2000 and more nodes.
TS_060	<b>Distances</b> – The GMT must be able to cover the distance on the GSI and FAIR site ( $\leq 2$ km).
TS_070	<b>Robustness</b> – The distribution of messages via the GMT shall be robust. The loss-rate of timing-messages should not exceed $10^{-12}$ , which corresponds to approximately one lost message per year. The failures due to losses of timing-messages must be negligible compared to other faults of the accelerator control system.
TS_080	<b>Determinism</b> – The GMT must transmit time critical messages with upper bound latency from the timing master to any node in the timing network.
TS_090	<b>Redundancy</b> – The timing network shall be implemented with redundant equipment (cables, switches). Redundancy is not required for connecting FTRNs to the timing network.
TS_100	<b>Campus-Wide Distribution</b> – The GMT shall support campus-wide distribution of time stamps, timing-messages and other machine or beam related telegrams. The distribution of information should not be limited to the accelerator control system.
TS_110	<b>FAIR Timing Receiver Nodes</b> – Requirements to the FTRNs are described in their detailed specification [7]. The FTRN requirements must be fulfilled by the GMT as a complete system. <ul style="list-style-type: none"> <li>• FTRN in-scope requirements shall be supported.</li> <li>• FTRN out-of-scope requirements must be implemented (TR_020, parts of TR_030, TR_070, parts of TR_170, TR_180, TR_250, TR_260, TR_270, TR_330, TR_380).</li> </ul>
TS_120	<b>Messages</b> – The GMT must be capable of transmitting prioritized messages in the timing network (QoS concept). The most common message types are <ul style="list-style-type: none"> <li>• Timing-Messages for synchronizing actions. Announce-Messages for pre-triggering Front-End actions are time-critical and have the same format as Timing-Messages.</li> <li>• Info-Messages for transmission of float-values (beam-intensity...).</li> <li>• Action-Messages. Unless indicated, action-messages are processed by the front-end software. Examples are <ul style="list-style-type: none"> <li>○ <i>Commit Events</i> for switching between FEC registers with different set values.</li> <li>○ <i>Rollback Events</i> for reverting set values</li> <li>○ <i>Post-Mortem Events</i> for freezing or unfreezing ring buffers</li> <li>○ <i>Reset Events</i> for resetting a FEC. Reset events must be processed by the gateway/firmware of the FTRN.</li> <li>○ <i>Trigger-Events</i> for the Software Oscilloscope (see below).</li> </ul> </li> </ul>
TS_130	<b>BuTiS</b> – The GMT and BuTiS are connected. The grandmaster clock of the GMT shall be derived from BuTiS clocks. Timestamps distributed by

	the GMT shall be phase locked to BuTiS clocks. The GMT shall deliver clock signals similar to the BuTiS T0 and c2 clocks on any FTRN.
TS_140	<b>GPSDO</b> – The GMT is connected to a GPSDO system. The GPSDO system is intended as a backup when a connection to BuTiS is not available or the GMT shall operate independently. In this case, all clocks and timestamps used and distributed by the GMT shall be locked to the GPSDO system.
TS_150	<b>RF-Systems DDS cards</b> – The RF-systems use DDS cards for generating ramps. The GMT must support the timely supply of these DDS cards with ramp data by indicating the allowed time slot for data transfer via the SCU backplane [22].
TS_160	<b>Parameterized Beam</b> – The GMT should support the concept of Multi Parameter Beam (also known as EFI), which was used during the operation of the SIS18 accelerator for cancer therapy. <sup>1</sup>
TS_170	<b>Info Telegram Service</b> – The GMT shall support distribution of important information, such as characterization of ions (nuclide, charge) or beam (intensity, focus). This information is obtained from the dedicated gateway <i>Facility Info Telegram Service</i> .
TS_180	<b>Log</b> – The GMT shall support logging important messages distributed in the timing network, such as timing-messages. The log shall include timing-messages that have been sent from the data master to the timing network.
TS_190	<b>Accelerator Shutdown GMT Operation</b> – The GMT shall be able to generate and distribute messages when the accelerator does not produce beam, e.g. outside beam times. This is required for testing components of the control system, equipment or experiments when no beam is produced. Messages must be distributed from the timing master, local generation at the FTRNs is not sufficient.
TS_200	<b>Info Message Gateway</b> – The GMT shall support the possibility to distribute messages via the general purpose network. This will be done via the dedicated <i>Info Message Gateway</i> .
TS_210	<b>User-Events</b> – The GMT shall support user-events, e.g. the possibility of a user-defined action with a configurable delay. The front-end software shall be able to assign a name (like “Hugo”) to such a user-event. N.B. This requirement is mainly mentioned for completeness but out-of-scope. It is fulfilled by combining the requirements TR_110 [7] and FSF_360 [15].
TS_220	<b>Local Accelerator Control Request Interfaces</b> – LACRITs are FTRNs which provide I/O to users (e.g. beam request, waveform generation). <ul style="list-style-type: none"> <li>• LACRITs are managed by the accelerator control system.</li> <li>• The GMT shall support configuring these interfaces via the timing network.</li> <li>• The GMT must be able to change the schedule according to the requests.</li> </ul>
TS_230	<b>Existing Machines and Control System Integration</b> – The GMT shall integrate/modify/replace the timing system of the existing machines and control system. Requirements are described in [11]. The GMT should link

<sup>1</sup> Treatment of patients is no longer foreseen.

	<p>to the existing timing generation of the UNILAC.</p> <ul style="list-style-type: none"> <li>• The GMT shall replace the timing generation for the SIS18 and the ESR.</li> <li>• The GMT should replace the timing generation of the transfer channel.</li> </ul>
TS_240	<b>p-Linac</b> – The GMT shall include the timing generation of the p-Linac, which is an injector to the SIS18 via parts of the existing transfer channel.
TS_250	<b>Settings Management System</b> – The GMT shall execute the schedule or alternative execution sequences that are generated by the SM. It must be possible to change parts of the schedule or the whole schedule during operation. Details on the requirement of the SM to the GMT are described in [3].
TS_260	<b>On-the-Fly Schedule Adjustment</b> – The GMT shall support adjusting the given schedule depending on input or feedback from other equipment. This functionality must be provided for specific exceptional cases like bunch-to-bucket transfer between ring machines. In such a case, the duration of involved beam processes is altered by the data master.
TS_270	<b>Online Schedule Monitor</b> – The GMT shall provide an on-line status of its activities to the OSM [12]. This includes the actual execution status of cycles, beam production chains, beam patterns, executed alternatives, relevant interlock information and status of beam requests.
TS_280	<b>Time Server</b> – The GMT shall provide a time server (e.g. NTP) and must synchronize it.
TS_290	<b>Timing Master FEC Software</b> – The GMT shall provide a FESA class representation of the timing master.
TS_300	<b>Timing Master FEC API</b> – The GMT shall provide a library API that provides the communication between the timing master and its FESA class (FEC software framework [15]). The API shall include the functionality of translating the schedule requested by the SM to the internal format required by timing master.
TS_310	<b>Alarm System</b> – The GMT shall link to the alarm system via the implementation of its FESA class [16].
TS_320	<b>Archiving System</b> – The timing master’s FESA class must provide properties for archiving relevant data by the archiving system [17].
TS_330	<b>Diagnostic Logging System</b> – The GMT shall generate and log text entries for relevant information via its FESA class [18].
TS_340	<b>Soft Oscilloscope System</b> – The GMT shall provide timestamps to the software oscilloscope system (SOS) [19]. The GMT may support the SOS by distributing trigger events to the SOS.
TS_350	<b>Post Mortem System</b> – The GMT must support the PM [8]. Specifically it must distribute the post mortem triggers for freezing und unfreezing FECs. Details on the requirements of the PM on the GMT are given in [8]. The GMT must fulfill the requirements PM_010, PM_020, PM_030, PM_060 and PM_210.
TS_360	<b>Interlock System</b> – The GMT must support the IL [9]. The data master as central scheduler must react on interlocks according to dynamically programmed decision trees provided by the SM. The GMT shall be able to react on interlocks at decision points predefined by the SM.
TS_370	<b>Return-Channel</b> – The GMT should provide a return-channel for bidirectional transmission of data. The use of the return-channel must be

	approved on with the GMT. It is reserved for well justified cases and must not disturb the topdown transmission of mission critical timing-messages.
TS_380	<b>Peer-to-Peer</b> – The GMT may provide means of peer-to-peer communication. Such communication must be approved by the GMT. It is reserved for well justified cases and must not disturb the topdown transmission of mission critical timing-messages.
TS_385	<b>VLAN</b> – The GMT should provide VLAN capability. This is not intended for the accelerator control system but for experiments which require local timing-messages for their control systems and a connection to the FAIR clock and data master. VLANs exist only locally and user messages must never be transmitted via the timing network of the accelerator.

**Table 1:** List of Functional Requirements

### 2.3.2 Non-functional Requirements

The GMT fulfills the following non-functional requirements:

Number	Description of the Requirement
TS_390	<b>Jitter</b> – The jitter of clock signals or output signals on FTRNs should be 100 ps or less. In any case, the jitter must be 500 ps or less.
TS_400	<b>Precision</b> – The precision of clock signals or output signals should be 1ns or better for FTRNs connected via optical links. For most of the FECs that are triggered by interrupt on the host bus system, a precision of 1 $\mu$ s shall be achieved.
TS_410	<b>Accuracy</b> – The accuracy of clock signals derived from the BuTiS T0 clock shall be sufficient for an unambiguous identification of c2 (200 MHz) clock cycles. This shall be achieved on FTRNs connected via optical links.
TS_420	<b>Granularity</b> – The granularity for signal detection or signal output shall be 8 ns or better.
TS_430	<b>BuTiS Clocks</b> – If connected via optical links, FTRNs shall be able to re-synthesize a signal with frequency and phase of the BuTiS T0 clock and an accuracy sufficient for identification of c2 clock cycles. FTRNs shall be able to re-synthesize a clock signal derived from the BuTiS c2 with the correct frequency.
TS_440	<b>Latency</b> – The upper bound latency for the transmission of timing-messages should be 200 $\mu$ s or less.
TS_450	<b>clockClass</b> – The GMT should implement clockClass 6 according to IEEE 1588-2008.
TS_460	<b>Appearance</b> – The timing master and its infrastructure is a showpiece to visitors and shall be an eye-catcher.
TS_470	<b>Local Delay Compensation</b> – FTRNs provide real-time reliability by correcting for local delays, see TR_330. The timing system defines a system-wide and unique upper limit for delays that can be compensated locally. The value for the upper limit will be defined later and its value is expected to be in the order of several $\mu$ s. Remark: Local delays exceeding the upper limit may be handled using Announce-Messages.

**Table 2:** List of Non-functional Requirements

The GMT shall follow the *basic requirements for the control system conception* as described in the common specifications [1].

### **2.3.3 General Constraints**

The GMT is based on the White Rabbit standard [14].

### **2.3.4 Architectural Principles**

The GMT shall respect the architecture and principles as described in the common specifications [1].

## **3 Technical Specifications**

From the point of view of the front-end software, the interface to the timing system is mainly the API Timing Library (ATL) as described in [7]. The front-end software uses the ATL to configure an FTRN such that the front-end software receives a timing event from the FTRN at the very moment when a real-time action shall be executed. Typically, this involves notifying the front-end software by an interrupt issued by an FTRN. Subsequently to the interrupt, the front-end software reads the data of the timing event through the host system bus. The data includes information such as event number and beam process ID.

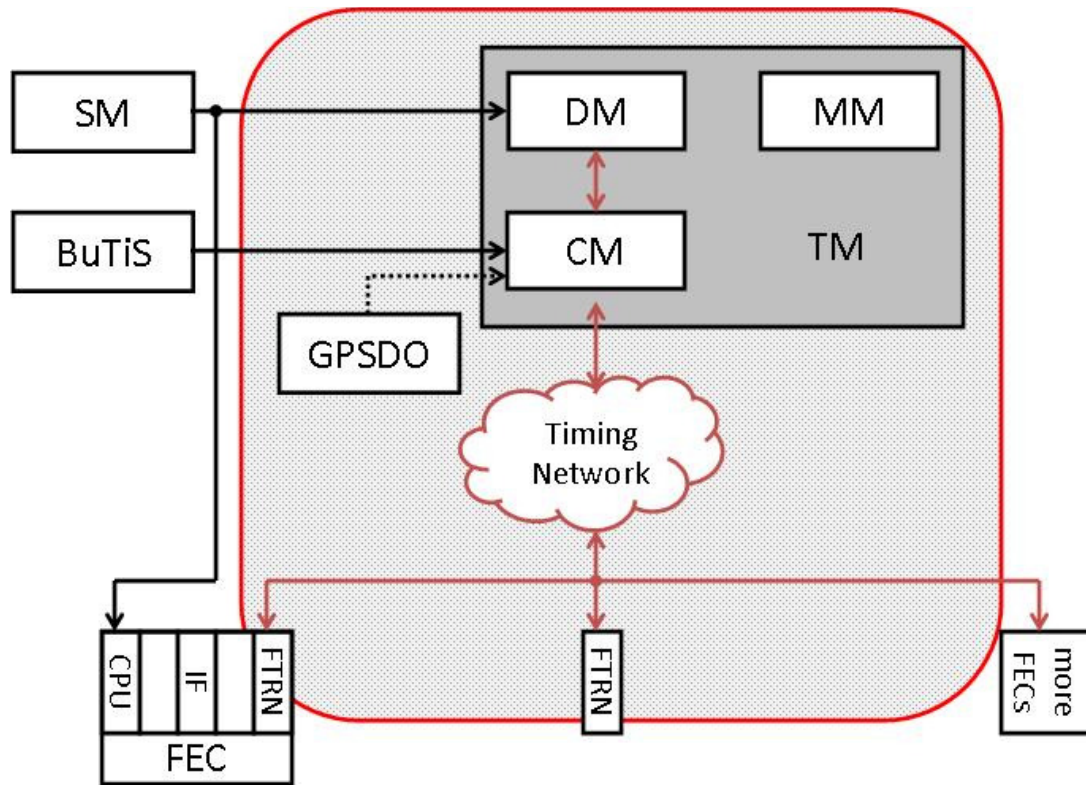
This section does not consider the point of view of the front-end software as described in the previous paragraph. Instead the technical specifications are described from the point of view of the timing system.

The GMT is an application of White Rabbit (WR). WR is a protocol being developed by CERN, GSI and other partners for synchronizing nodes in a packet-based network. It combines Gigabit-Ethernet, IEEE1588-2008 (PTP), precise knowledge of the link delay and Synchronous Ethernet: Time synchronization is achieved by adjusting the clock phase (125 MHz carrier) and offset (Coordinated Universal Time – UTC, or International Atomic Time – TAI) of all network nodes to that of a common grandmaster clock. It has been demonstrated in 2012, that sub-nanosecond synchronization with a jitter in the hundreds of picoseconds range is achieved over distances of a few kilometers and across dedicated WR network switches.

Equipment action will not be directly triggered upon receiving signals from a central timing unit. Instead, timing systems using WR networks are based on the notion of absolute time. Timing messages are sent from the timing master to the timing receivers and include an identifier to an action and the time of action execution. When a relevant timing message is received by a timing receiver, an action is scheduled for execution at a given time and date. Like an alarm clock, a timing receiver executes the action autonomously on time. Thus, distribution of information through the network and timely execution of actions are decoupled. Typically, the action executed by the timing receiver is configured by the front-end software via the ATL.

### **3.1 The General Machine Timing System**

This section indicates the proposed solution path for the implementation of the GMT. The proposed solution is based on the requirements.



**Figure 4:** Sketch of the GMT (see text).

Figure 4 shows a sketch of the GMT and its most important sub-systems. The GMT consists of the *Timing Master* (TM) and the *timing network* which connects to the receiver nodes (FTRN). The FTRNs are integrated into front-end Controllers (FECs). Connected to the timing master is the *settings management* (SM) and the high precision clock distribution system BuTiS backed up by a GPSDO.

The timing master contains the *data master* (DM), the *clock master* (CM) and the *management master* (MM).

**Data Master.** The data master is the “heart” of the GMT and provides the real-time scheduler of the accelerator control system.

**Clock Master.** The clock master is a dedicated White Rabbit switch. It is the topmost switch layer of the timing network and provides the grandmaster clock which is distributed to all other nodes in the timing network. The clock master derives its clock and timestamps from the highly stable BuTiS clocks or a primary reference time source.

**Management Master.** All active components including receiver nodes and switches are registered to the management master. The management master has the task of monitoring and managing the active components of the timing system.

From a simplified point of view of the GMT, the accelerator control system works as follows. The settings management provides set values to devices of the accelerator control system. In the case of the data master this is the schedule of the accelerator complex. In the case of a FEC controlling a magnet, this could be the data of a ramp. The FEC then configures its local FTRN to react on relevant *timing-messages*. A timing message contains identifiers including beam production chain ID, beam process ID, event number, and the time of execution. During beam production, the real-time scheduler inside the data master broadcasts timing messages to

the timing network. After receiving such a timing message<sup>2</sup>, the FTRN waits until the time of execution is due and then triggers the ramp start by sending an interrupt to its FEC.

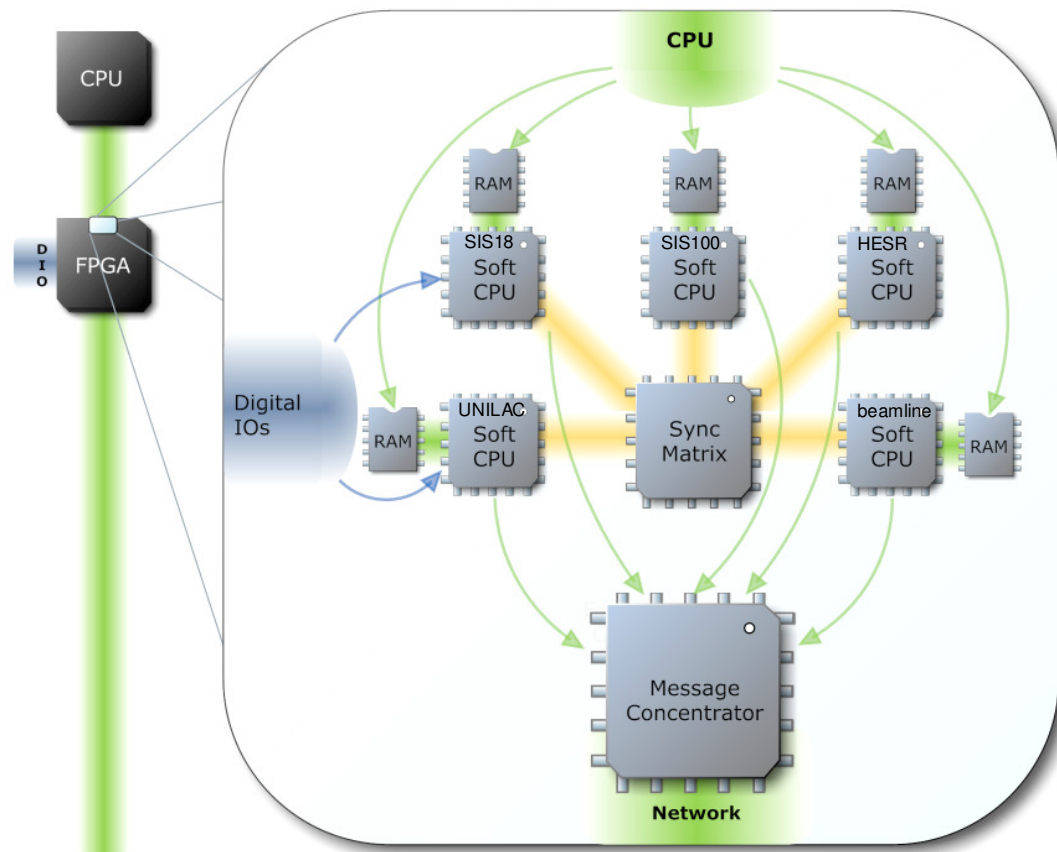
From the point of view of the settings management, the timing master and the FEC represent devices in the equipment layer of the accelerator control system architecture (see Figure 1). As the FECs, the timing master is represented by its FESA class [15].

### 3.2 GMT Components

The main components of the timing system are the master, the network and the receiver nodes.

Hardware components shall follow the corresponding guidelines [1], [23].

#### 3.2.1 Timing Master



**Figure 5:** Usage of Soft-CPUs in the data master.

The timing master is the central component of the timing system. It has many interfaces to other components of the accelerator control system as described in section 2.2.2. Of prime interest here is the data master. Its components include a CPU and a field programmable gate array (FPGA) as sketched in Figure 5. The CPU runs the front-end software, which is based on the Front-End System Architecture (FESA). The FPGA includes several Soft-CPUs, each of those is dedicated to a specific machine or timing section. The task of each Soft-CPU is

<sup>2</sup> Timing messages are broadcasted by the data master. FTRNs implement a filter and react only on relevant timing messages.

generating timing messages for the super cycle of its machine. As the real machines require synchronization for executing beam production chains, the generation of timing messages by the Soft-CPU's requires synchronization. This is achieved by a *sync matrix*. The timing-messages generated by the Soft-CPU's are bundled and sent to the physical network by the *message concentrator*.

The main interface of the timing master is the one from the data master to the settings management system (SM) via the front-end software. The SM generates set values and transmits them in an appropriate format like a decision tree or a domain specific language to the DM. The FESA class of the DM uses a library for on-the-fly

- source-code generation,
- cross-compilation of Soft-CPU codes and
- uploading codes to Soft-CPU's.

The idea behind this procedure is the following. Instead of using a complex code including all use-cases and operational modes of the accelerator, the Soft-CPU's only run simple code that just implements the requested operational mode and schedule of the accelerator. The result is fairly simple machine specific Soft-CPU code. Of course, the codes include alternatives to the schedule, which are executed depending on conditions imposed by the interlock system or beam requests, see section 2.2.2. More information on the timing master is given in [20].

### 3.2.2 Timing Network

The timing network consists of the infrastructure required for distributing clock signals and timestamps from the clock master to all nodes as well as transmitting data between nodes. The active components of the timing network are mostly dedicated switches supporting the White Rabbit protocol. The timing network is monitored. All nodes implement SNMP. The management features of the WR switches are accessible via dedicated management ports.

The three main features of the timing network are redundancy, robustness and determinism. More information is given in [21].

**Redundancy.** The timing network shall provide redundant paths between the topmost and the bottommost switch layers. Redundant links should be provided between the data master and the topmost switch layer. The topmost switch layer should be implemented as the clock master within the timing master. There are no redundant links between the bottommost switch layers and the timing receiver nodes.

**Determinism.** The timing network shall provide deterministic transmission of time critical data from the data master to the timing receiver nodes. Transmission of such data must be guaranteed with upper bound latency from the data master to the nodes. This requires an appropriate implementation of the firmware of the White Rabbit switches and may include features such as a cut-through path or adequate scheduling of queues.

**Robustness.** Upper bound latency in the order of 200  $\mu$ s requires transmission protocols without hand-shake or retransmission such as UDP or raw Ethernet. Thus, the data master just sends the data but has no knowledge of their successful transmission. The timing network must provide sufficient robustness, in order to ensure that the probability of losing data during transmission, followed by possible subsequent damage to the machine, is small compared to other possible failures in the accelerator control system. Robustness is achieved by using forward error correction techniques. While linear codes may be applied to correct bit errors, other codes like Reed Solomon codes are used to compensate for loss of packets in the network.



### 3.2.3 Timing Receiver Nodes

Specifications on the timing receiver nodes are given in [7].

### 3.2.4 MAC and IP Addresses

Each node in the timing system has a unique vendor specific MAC address. Private MAC addresses must be avoided, such that it is possible to use a node manufactured by GSI at CERN and hopefully vice versa. Nodes provided by GSI use a GSI OUI as part of the MAC address. Nodes store their MAC address in non-volatile RAM such as an EEPROM. All manufacturers of WR nodes used at FAIR are responsible for delivering nodes with MAC addresses and make the MAC addresses for each device known to the GSI timing team. WR nodes obtain their IP addresses via BOOTP after power-on. The GSI timing team has the task of maintaining a list of WR nodes, their MAC and IP addresses.

A BOOTP server is part of the management master and shall provide IP addresses to WR nodes in the timing network. It is the responsibility of the GSI Timing Team to maintain the assignment of IP and MAC addresses.

### 3.2.5 Announce Messages and Local Delay Compensation

Every action at a FEC is executed with a specific delay. Delays in the nanosecond range may be caused by the routes in a PCB or cables. Delays in the microsecond range may be caused by the connected equipment or, in case of a FESA real-time action, may depend on the latency of the host system interrupt lines, CPU speed or operating system. Some equipment, like power supplies for charging capacitors, may have a delay of seconds. The timing system follows two different strategies. In case the delay is in the order of nanoseconds to microseconds, the concept of local delay compensation is used and delays are compensated autonomously by the timing receivers. Longer delays are handled by sending Announce-Messages via the Data Master. Both concepts are described in the following.

As described in TR\_330, FTRNs provide real-time reliability by taking account local delays [7]. FTRNs compensate such delays, by applying a pre-trigger value to the timestamp of the requested action. The length of a pre-trigger must not exceed the upper limit, which is defined by the timing system, see TS\_470. Local delay compensation is configured by the Front-End software through the Timing API Library. Local delay compensation avoids the necessity of sending “Announce Events” in case the local delays are short compared to the upper bound latency of the timing network.

The Settings Management must be aware of the local delays for the requested actions in the FECs. In case the requested action requires a compensation exceeding the upper limit for local delay compensation, the Settings Management must add “Announce Messages” to the schedule that it provides to the Data Master.

Propositions.

1. The upper limit for local delays is “fixed” and stored in a database. Its value should be in the order of a few or tens of microseconds and smaller than the upper bound latency of the timing network. The Data Master takes this into account by adding the upper limit for local delay compensation (TS\_470) to the value of the latency of the timing network (TS\_440).
2. The upper limit for local delays is set at the FECs via a FESA property via the Settings Management.

3. The local delays required for specific actions at specific FECs are stored in a database and used by the Settings Management to prepare the schedule. The Settings Management then complements the schedule by “Announce Messages”, in case the local delays exceed the limit for local delay compensation.
4. In case an “Announce Message” is required, the FEC (and by this it’s FTRN) is programmed to trigger at the “Announce Message”.

### **3.2.6 GMT Supporting Systems**

**GPSDO.** In case the GMT operates independently of the BuTiS centre, it should continue operation with clockClass 6 according to IEEE 1588-2008. This requires a primary reference time source such as a GPS disciplined oscillator. The internal stability of the GPSDO should be sufficient for an operation of the GMT even in case GPS signals are jammed or not available. A hand-over mechanism from the BuTiS centre to the GPSDO will be implemented.

**Time Server.** The Time Server provided by the GMT shall be visible throughout the accelerator control system network. It shall implement a protocol that is common in the environment of the accelerator control system such as NTP.

**Info Message Gateway.** The Info Message Gateway shall implement the functionality of a gateway between the GMT and accelerator control system network.

**Local Accelerator Control Request Interfaces.** This type of equipment is typically used in local areas of FAIR experiments. They serve as interface between the GMT and the experiments and are part of the GMT.

## 4 Acronyms and Definitions

The following acronyms and definitions are used in this document.

- **GSI Timing Team:** The Timing Group within the Controls department (BEL) at the GSI Helmholtz Centre for Heavy Ion Research GmbH (GSI) Darmstadt.
- **Firmware:** compiled software code. Firmware is loaded into a Soft-CPU embedded in the gateway.
- **Gateway:** synthesized HDL code. Gateway is loaded into the FPGA of a timing node, such as an FTRN. In many cases, gateway includes a Soft-CPU such as the lm32.
- **Clock Master:** grandmaster clock of the GMT, source of timestamps and clocks within the GMT.
- **Data Master:** the “heart” of the GMT, provides the real-time scheduler of the accelerator control system.
- **Grandmaster Clock:** The root timing reference in a PTP network.
- **Management Master:** manages and monitors the timing system
- **Timing Event:** This is a term used by the front-end software FESA. A timing event triggers a real-time action in the software. From the point of view of the GMT, this is called an *action*.
- **Timing Master:** includes data master, clock master und management master
- **Timing Message:** Data distributed via the timing network relevant for the timing of the FAIR accelerator complex. A timing message includes the process ID of a specific production chain and the time of execution. From the point of view of a FTRN, a timing message is an *event* for the ECA unit [7].
- **Timing Network:** dedicated network connecting the nodes of the GMT
- **BuTiS** - Bunch Phase Timing System
- **FEC** - Front-End Controller
- **FESA** - Front-End System Architecture
- **FPGA** - Field Programmable Gate Array
- **FTRN** - FAIR Timing Receiver Node
- **GMT** - General Machine Timing system
- **GMTS** - GMT Supporting system
- **GPS** - Global Positioning System
- **GPSDO** - GPS Disciplined Oscillator
- **IL** - Interlock System
- **LACRITs** - Local Accelerator Control Request Interfaces
- **NTP** - Network Time Protocol
- **OSM** - Online Schedule Monitor
- **PM** - Post Mortem system
- **SCU** - Scalable Control Unit
- **SM** - Settings Management system
- **SOS** - Soft Oscilloscope System
- **TM** - Timing Master
- **WR** - White Rabbit

## I. Related Documentation

- [1] F-CS-C-01e, FAIR Common Specification “Accelerator Control System”
- [2] MIL Timing System of the Existing Facility, Controls Department, private communication
- [3] F-DS-C-03e, FAIR Detailed Specification “Settings Management System”
- [4] F-DS-C-31e, FAIR Detailed Specification “Machine protection systems”
- [5] F-DS-C-20e, FAIR Detailed Specification “Personnel Safety System”
- [6] F-DS-C-02e, FAIR Detailed Specification “Equipment Interface and Control”
- [7] F-DS-C-06e, FAIR Detailed Specification “Timing Receivers”
- [8] F-DS-C-13e, FAIR Detailed Specification “Post Mortem System”
- [9] F-DS-C-08e, FAIR Detailed Specification “Interlock System”
- [10] F-CS-RF-14e, FAIR Common Specification “Bunch Phase Timing System”
- [11] F-DS-C-29e, FAIR Detailed Specification “Existing Machines and Control System Integration”
- [12] F-DS-C-04e, FAIR Detailed Specification “User Application Programs”
- [13] Tomasz Włostowski, “Precise time and frequency transfer in a White Rabbit network” Warsaw University of Technology, May 2011.
- [14] E. G. Cota et al., “White Rabbit Specification: Draft for Comments” (2011) [www.ohwr.org](http://www.ohwr.org).
- [15] F-DS-C-01e, FAIR Detailed Specification “FEC Software Framework”
- [16] F-DS-C-09e, FAIR Detailed Specification “Alarm System”
- [17] F-DS-C-11e, FAIR Detailed Specification “Archiving System”
- [18] F-DS-C-10e, FAIR Detailed Specification “Diagnostic Logging System”
- [19] F-DS-C-14e, FAIR Detailed Specification “Soft Oscilloscope System”
- [20] Mathias Kreider, “The FAIR Timing Master: A Discussion of Performance Requirements and Architectures for a High-Precision Timing System”, Proc. ICALEPCS 2011, Grenoble, France.
- [21] Maciej Lipinski et al., “Reliability in a White Rabbit Network”, Proc. ICALEPCS 2011, Grenoble, France.
- [22] Tibor Fleck, FAIR Technical Note no. 20100510, 2010.
- [23] F-TG-ET-01e, FAIR Technical Guideline “Electrical Design Rules and Regulations”.
- [24] F-DS-C-05e, FAIR Detailed Specification “General Machine Timing System”

## II. Document Information

- 21 January 2013: distilled from the detailed specifications of the General Machine Timing System, F-DS-C-05e.