

## DOCUMENT

Document /  
Deliverable Name: **Analysis and prioritization of ex-  
ploitable cross-domain services  
with focus on the railway domain**

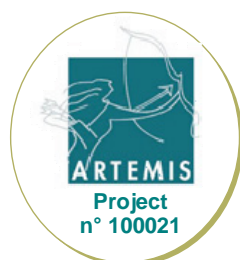
Document /  
Deliverable Nr. **D 4.1**

Version:  **draft** **1.0**  
 **final**

Document Type:  **confidential**  
 **public**

Responsible: **Constantin Sârbu, TUDA**

Date of creation:: **30 December 2009**  
Last modification **30 December 2009**





## List of INDEXYS Beneficiaries

No	Name	Short	Country
01	TTTech Computertechnik AG	TTT	Austria
02	AUDI AG	AUDI	Germany
03	Delft University of Technology	DUT	Netherlands
04	EADS Deutschland GmbH	EADS-IW	Germany
05	NXP Semiconductors Netherlands B.V.	NXP-NL	Netherlands
06	OptXware Research and Development Ltd.	OPT	Hungary
07	Thales Rail Signaling Solutions GmbH	TRSS-AT	Austria
08	Technical University of Darmstadt	TUDA	Germany
09	Technical University of Kaiserslautern	UNIKL	Germany
10	Vienna University of Technology	TUWI	Austria

## Author(s)

Name	Company
Chr. Scherrer	TRSS-AT
Chr. Fidi	TTT
C. Sârbu	TUDA

## Project Coordination

### TTTech Computertechnik AG

Schoenbrunner Strasse 7  
1040 Vienna, Austria

#### Technical Matters:

D.I. Andreas ECKEL, MBA  
Email: andreas.eckel@tttech.com  
Tel: +43 1 585 34 34 – 16  
Fax: +43 1 585 34 34 – 90

#### Financial Matters:

D.I. Andreas BAUMGARTNER  
Email: andreas.baumgartner@tttech.com  
Tel: +43 1 585 34 34 – 942  
Fax: +43 1 585 34 34 – 90

Copyright 2009: The INDEXYS Consortium  
[www.indexys.eu](http://www.indexys.eu)

## Revision chart and history log

Version	Date	Reason
0.1	04/09/2009	Initial version
0.2	07/10/2009	Adding Features Section
0.3	16/11/2009	Included WP4 partner inputs and comments; added section 3.2 and 5.3; multiple formatting issues.
1.0	30/12/2009	Final version - Minor changes, fixed references, typos

## Table of Contents

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>7</b>
<b>2.</b>	<b>FAULT TOLERANT PLATFORMS IN THE RAILWAY DOMAIN .....</b>	<b>8</b>
2.1.	TAS Platform Architecture Overview .....	8
2.1.1.	Inter Replica Communication System.....	9
2.1.2.	Fault tolerance system.....	9
2.1.3.	Communication interfaces .....	10
<b>3.</b>	<b>TAS PLATFORM IN THE GENESYS CONTEXT.....</b>	<b>12</b>
3.1.	Mapping of architectural features of Genesys onto the TAS Platform.....	12
3.2.	Mapping of Genesys Services onto the TAS Platform .....	14
3.2.1.	Genesys Core Services .....	14
3.2.1.1.	Basic Configuration Services.....	15
3.2.1.2.	Basic Execution Control Services.....	15
3.2.1.3.	Basic Time Services .....	16
3.2.1.4.	Basic Communication Services .....	16
3.1.2 .	Genesys Optional Services.....	17
3.2.1.5.	Diagnostic Services .....	17
3.2.1.6.	External Memory Management Services.....	18
3.2.1.7.	Security Services .....	18
3.2.1.8.	Resource Management Services .....	19
3.2.1.9.	Gateway Services.....	19
3.2.1.10.	Mobility Services.....	20
3.2.1.11.	Generic Middleware Services .....	21
<b>4.</b>	<b>REQUIREMENTS .....</b>	<b>22</b>
4.1.	Error containment regions.....	22
4.2.	Communication .....	23
<b>5.</b>	<b>FEATURES OF DOMAIN SPECIFIC PLATFORM SOLUTION .....</b>	<b>29</b>
5.1.	TTEthernet Introduction .....	29
5.2.	Software-Based TTEthernet .....	30
5.3.	Scheduling Mechanisms for TTEthernet.....	31
5.4.	Mapping of TTEthernet features to the TAS Platform requirements .....	32
<b>6.</b>	<b>REFERENCES .....</b>	<b>38</b>

## List of Figures

Figure 1: TAS Platform Component/System Architecture .....	8
Figure 2: Interaction of standards.....	29
Figure 3: Software-Based TTEthernet detailed architecture .....	30
Figure 4: Software-Based TTEthernet with Operating System Driver .....	31

# 1. Introduction

The aim of the INDEXYS project is the implementation of the principles, concepts, and methodology defined in the GENESYS project. The target domains of the project are automotive, railway, and aerospace; therefore the implementation should be focused on legacy, closed systems that are present in these areas.

In the current document we will analyze architectures in the railway domain and map requirements to the GENESYS architectural style. Both the requirements and the features of the railway domain are analysed in the context of the core and optional services of GENESYS, as defined in [GENESYS].

The document is organized as follows: Chapter 2 introduces TAS Platform as a fault tolerant architecture in the railway domain. Chapter 3 maps TAS Platform architectural principles to the GENESYS architectural style and services as it is presented in [INDEXYS\_D1.1]. From this analysis, requirements on the GENESYS architectural style are derived from a railway domain perspective in chapter 4. Chapter 5 presents the features of the railway domain-specific platform solution.

## 2. Fault Tolerant Platforms in the Railway Domain

This chapter introduces fault tolerant platforms in the railway domain. The introduction of generic fault tolerant computing platforms in the railway domain is driven by the large contrast in lifetime of signaling applications as compared to the quick pace basic technology as well as hardware evolves. To keep pace with this rapid technology evolution, the TAS Control Platform separates the rail applications from the hardware and software technologies, ensuring that complex railway applications have a long useful life by implementing a stable interface.

The TAS Control Platform is an open, scalable software architecture oriented towards established industrial computing standards. Its core incorporates software components such as a Portable Operating System Interface (POSIX) compliant operating system, a fault tolerance system and a communication system. The fault tolerance system offers various configurations for actively redundant systems, ranging from redundant 2 out of 2 and 2 out of 3 systems to non-redundant 1 out of 1 systems. The communication system offers a number of standard communication services, such as Internet Protocol (TCP/IP family), serial lines and field buses (CAN controller area network, TTP time triggered protocol, PROFIBUS Process Filed Bus), as well as specific safe communication services conforming to European Committee for Electro technical Standardization (CENELEC) standards [EN\_50129].

At the hardware level, the TAS Control Platform uses commercial off-the-shelf components, which are supplemented by added-value services for railway control systems.

### 2.1. TAS Platform Architecture Overview

Figure 1 shows the component architecture of TAS Platform. A computing node (CN) is the logical target computer. It may consist of 1 up to 3 individual computing elements (CEs), depending on the

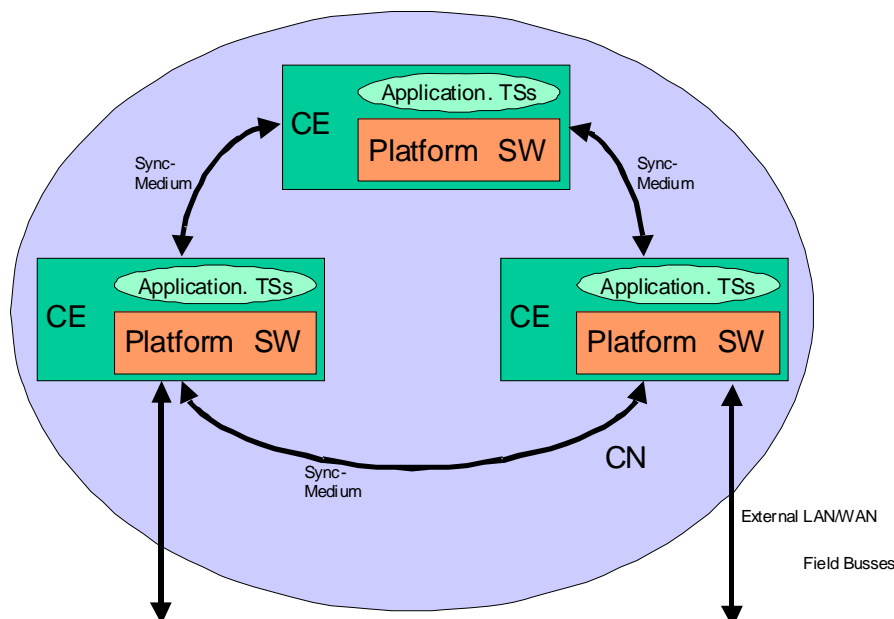


Figure 1: TAS Platform Component/System Architecture

application systems replication degree. A computing element (CE) refers to a physical computer that is synchronized with other CEs of the same CN. A Task set (TS) is a set of tasks forming a logical application software entity.

The synchronization medium serves inter replica synchronization and is implemented as point-to-point network based on Ethernet.

## 2.1.1. Inter Replica Communication System

The TAS control platform redundancy mechanisms are based on active replication. Replicated task-sets are said to be deterministic [POLE96] if, in the absence of faults, any execution of replicas starting from the same initial state and consuming the same ordered set of input messages leads to the same ordered set of output messages. In real-time systems, the same ordered set of output messages has to be produced within a given time interval.

The communication system is a run-time environment that supports replica determinism of actively redundant applications. Communication between these replica deterministic applications is realized solely via POSIX message queues. To this end, POSIX message queue system calls and POSIX timing system calls (15 system calls in all) are wrapped and supported by the communication system. All other POSIX system calls are provided by the operating system. The communication system provides these POSIX real-time services with extended semantics:

- **Replica transparency and Redundancy handling:** The application knows nothing about its own redundancy or about the redundancy of its receivers or senders. Replication and fault tolerance are transparent to the application.
- **Location transparency:** Transmission of messages is not bound to a local computer at the application level.
- **Authorization:** To supervise communication in the system, authorization has to be explicitly defined in a static, off-line described configuration file. Message flow in the system is then checked against this configuration file.
- **Replica deterministic time base:** As the local clock cannot be used for replica deterministic applications, the communication system provides a replica deterministic time base.
- **Application scheduling:** Different application task sets (sets of closely related processes) are executed asynchronously, with preemptive priority scheduling allowing interleaving of their executions on each replica. Within an application task set, run to completion scheduling is applied between pre-emption points.
- **Support of programming models:** To preserve replica determinism of redundant applications, some semantic restrictions of the full POSIX API apply. To reduce the complexity of handling these restrictions, the communication system supports programming models to ensure deterministic behavior of redundant applications.

## 2.1.2. Fault tolerance system

The platform fault tolerance system provides software-implemented fault tolerance of loosely coupled computers. Fault tolerance is achieved by comparing the message flow of actively redundant applications. All fault tolerance mechanisms are realized only in software, without the need for dedicated hardware components. To guarantee that fault-free applications perform the same

operations and behave replica deterministically, the applications have to follow the programming models supported by the communication system.

- **Configurable redundancy:** The fault tolerance system supports various configurable redundancy schemes at the computer and application levels. At the computer board level, it supports 2-out-of-3 and 2-out-of-2 configurations. In 1-out-of-1 configurations, used in application provided fault tolerance implementations an empty layer represents the fault tolerance system. On the redundant computer system, applications can execute using various redundancy schemes, ranging from 1-out-of-1 to 3-out-of-3. It should be noted that the redundancy configuration for the application task-sets need not be identical to that of the computing node (CN). For example, it is possible to run a 2-out-of-3 application together with a 1-out-of-1 application on a 2-out-of-3 computer system.
- **Message-based comparison:** Application data is forwarded via the communication system to the fault tolerance system. Data is consolidated across all replicas after an interactive consistency exchange via fault tolerance dedicated synchronization links. In the next step, the fault tolerance system votes on this data according to the configured redundancy scheme and delivers the voted result to the communication system for forwarding to the receiver application.
- **Message comparison method:** The fault tolerance system supports different comparison methods, such as majority decision, master-slave processing, byte-wise comparison and semantic comparison.
- **Delivery of replica deterministic time base:** The fault tolerance system provides the communication system with a synchronized replica deterministic time base.
- **Fault management:** As well as comparing redundant messages, the fault tolerance system gathers fault information from all subsystems and components, reports this information to a diagnostic system and coordinates the actions taken to resolve the problem (e.g. isolating an application, rebooting a computer).
- **On-line recovery:** In 2-out-of-3 redundancy schemes, fault management includes the reintegration of faulty components (applications and/or computers) during operation without service interruption. To achieve this, the state of the two remaining active components is transferred to the recovered component. Recovery units are applications and/or computers. In the event of a computer failure, the entire computer and all the applications executing on it are recovered. Recovery of an application means that only the failed application is recovered, without affecting other applications running on the computer.

### 2.1.3. Communication interfaces

TAS platform discerns between three kinds of communication interfaces:

- **Inter-Replica replica communication for synchronization and fault tolerance services:** The inter-replica communication links are designed as point-to-point links and are solely reserved for replica synchronization and fault tolerance as well as recovery services.
- **Local Interfaces:** TAS Platform is being used for a wide variety of safety-critical railway applications which all have their specific communication requirements. Thus a large spectrum of external buses is supported, such as CAN, TTP, Ethernet, MVB, PROFIBUS, to name a few.
- **External communication (Internet connectivity):** TAS Platform does not define a strict external communication model. The only restrictions defined by the TAS Platform communication model refers to redundancy and safety constraints, as a single replica cannot produce a safety critical message.

To provide the applications a service to enforce safety constraints according to the European standard EN 50159 ([EN\_50159\_1] and [EN\_50159\_2]) for external communication, the “One Channel Safe” (OCS) communication protocol is provided. OCS implements communication message voting on top of TAS Platform voting services, transmission error detection capabilities, re-transmission and redundancy link management.

## 3. TAS Platform in the GENESYS context

This section establishes the relations between Genesys architectural principles and defined services and the TAS platform for the railway domain in the context of the INDEXYS project. First, the Genesys principles are discussed in section 3.1 and then the mapping of Genesys services to the railway domain are listed in section 3.2. For reference, all the Genesys aspects are detailed in the GENESYS book [GENESYS].

### 3.1. Mapping of architectural features of Genesys onto the TAS Platform

This section compares architectural principles of the GENESYS architecture and the TAS Platform and follows the listing given in [INDEXYS\_D1.1, section 2.2]. For each GENESYS paradigm the relating TAS Platform realization is listed.

GENESYS Style	TAS Platform implementation
<b>Component-based design</b>	From the GENESYS architecture point of view a Computing Node (CE) is a component.
<b>Message-based communication infrastructure</b>	Inter-Replica Communication in TAS Platform is strictly message oriented. Additionally EN 50159-1 and its OCS implementation also define message based communication.
<b>Well defined component interfaces</b>	For inter-replica communication, TAS Platform uses LIFs in the GENESYS sense. For external communication, this is not the case.  TAS Platform does not define the properties of external links beyond safety application conditions. The requirements on external communication and local interfaces vary over application domains (on-board, Interlocking, trackside). In the GENESYS sense field bus communication interfaces can be seen as local interfaces. Local interfaces in TAS Platform are standard technologies (CAN,TTP, MVB, PROFIBUS, etc).
<i>Continued on next page →</i>	

<i>← Continued from previous page</i>	
<b>Multi-level structure</b>	Within the integration level definition of the GENESYS architectural style the railway domain only embraces the device level. From an architectural view, embedded computer systems including communication interfaces are the only relevant integration level to define.
<b>Networking management</b>	OCS provides networking management services (redundancy, reintegration, safety ...)
<b>Common time base</b>	Within a CN of TAS Platform a common time base is established and the local clocks are re-synchronized.
<b>Communication Modes</b>	Inter-replica communication can be time triggered as well as event triggered.  External communication and local interfaces: mostly event triggered (but MVB is time triggered)  Data streaming is not relevant in the railway domain
<b>Heterogeneous networks and internet connectivity</b>	This aspect is covered by the external communication interfaces (Ethernet, TCP, UDP)
<b>Integrated resource management</b>	Resource management in TAS Platform is governed by the operating system and the fault tolerance layer.
<b>Energy efficiency</b>	Energy efficiency in the railway domain is demanded by extreme environmental operating condition (power dissipation and temperature range) as well as overall product life cycle cost
<i>Continued on next page →</i>	

← Continued from previous page	
<b>Fault and error containment</b>	<p>Robustness services are meant to cope with physical faults as well as with design faults.</p> <p>With embedded computers being the only integration level being present in TAS Platform the fault containment regions are naturally given to be a single board computer, i.e. the CE. As being discussed in sect. 2.1 TAS Platform implements error containment regions with respect to physical faults by an active redundancy approach.</p>
<b>Integrated security</b>	<p>Security can be handled separately (external boxes), no integration into system architecture necessary</p>
<b>State awareness</b>	<p>TAS Platform implements recovery services to reintegrate failed CEs.</p>
<b>Diagnosis</b>	<p>Diagnostic services are being provided by the operating system and the fault tolerance and communication middleware.</p>

## 3.2. Mapping of Genesys Services onto the TAS Platform

Most of the GENESYS services are included in the TAS platform, as illustrated in the following tables. As expected given the generality of the GENESYS platform [GENESYS], almost all core services are provided - with the only exception of the real-time streaming service. Remarkably, most of the services needed to achieve consistency in a fault-tolerant manner are also included. Other services, such as wireless or internet connection, are not currently provided as the current TAS platform is a closed system but are foreseen to be integrated in the future.

Genesys distinguishes between two types of services, the “Core Services” and the “Optional Services”. Both types of services are detailed in the following sections, alongside with their sub-categories.

### 3.2.1. Genesys Core Services

Genesis core services fall into four distinct categories (configuration, execution control, time and communication services), listed in the next subsections.

### 3.2.1.1. Basic Configuration Services

Genesys defines three basic configuration services. The following table contains their mapping to the TAS platform.

Genesys Core Services (Basic Configuration Services) → TAS Platform				
Req.	Feature	Service Name	Important for Railway domain (yes/no)	Rationale for Importance / Implementation in TAS Platform
-	Rail:05	Identification Service	Yes	The TAS platform needs to be able to distinguish different physical units, i.e. computing elements (CEs)
-	-	Basic Boot Service	Yes	The TAS platform boots using fault tolerant communication system layer CS-FT
COM:06	-	Inter-Component Channel Configurator	Yes	Also done by the CS-FT configurator

### 3.2.1.2. Basic Execution Control Services

Genesys defines three basic execution control services. The following table contains their mapping to the TAS platform.

Genesys Core Services (Basic Execution Control Services) → TAS Platform				
Req.	Feature	Service Name	Important for Railway domain (yes/no)	Rationale for Importance / Implementation in TAS Platform
ECR:02	-	Start Execution	Yes	Execution control is done by the OS and its scheduler, which supports pre-emptive and run-to-completion scheduling
ECR:02	-	Terminate Execution	Yes	Execution control is done by the OS and its scheduler, which supports pre-emptive and run-to-completion scheduling
ECR:02	-	Reset	Yes	Execution control is done by the OS and its scheduler, which supports pre-

				emptive and run-to-completion scheduling
--	--	--	--	------------------------------------------

### 3.2.1.3. Basic Time Services

Genesys defines two basic time services. The following table contains their mapping to the TAS platform.

Genesys Core Services (Basic Time Services) → TAS Platform				
Req.	Feature	Service Name	Important for Railway domain (yes/no)	Rationale for Importance / Implementation in TAS Platform
COM:15	Rail:13	Common Time Service	Yes	Within a computing node (CN) of TAS Platform a common time base is established through a CS-FT synchronization layer, the CEs' time base is regularly synchronized on this global time base. There is no inter CN clock synchronization in the TAS platform.
-	-	Timer Interrupt Service	Yes	Preemptive scheduling allows for task interruption in the form of a context switch. Tasks wait in preemption points for a (time) interrupt to resume execution, which is controlled by the OS.

### 3.2.1.4. Basic Communication Services

Genesys defines three basic communication services. The following table contains their mapping to the TAS platform.

Genesys Core Services (Basic Communication Services) → TAS Platform				
Req.	Feature	Service Name	Important for Railway domain (yes/no)	Rationale for Importance / Implementation in TAS Platform
-	Rail:06	Periodic Exchange of Messages	Yes	The synchronization layer regularly exchanges messages within a CN to establish a global time base. Inter CN communication includes "Is alive" messages, which are exchanged periodically in case there is no other traffic.

COM:02 COM:03 COM:04	Rail:06	Sporadic Exchange of Messages	Yes	Data exchange between CNs is solely done through event triggered (sporadic) messages.
-	-	Primitive Real-Time Streaming	No	Data streaming is not important for the TAS platform.

### 3.1.2. Genesys Optional Services

Genesys optional services fall into seven distinct categories (diagnostic, memory management, security, resource management, gateway, mobility and middleware services), listed in the next subsections.

#### 3.2.1.5. Diagnostic Services

Genesys defines four optional diagnostic services. The following table contains their mapping to the TAS platform.

Genesys Optional Services (Diagnostic Services) → TAS Platform				
Req.	Feature	Service Name	Important for Railway domain (yes/no)	Rationale for Importance / Implementation in TAS Platform
-	-	State Externalization	No	There is no periodic state externalization in the TAS platform. Specific state parameters can be accessed on demand through a Unix-like /proc file system, which delivers state information on middleware, memory-usage, etc. This information is insufficient for recovery in GENESYS.
-	-	Membership Service	Yes	Provided by safety middleware CS-FT.
-	-	Analysis of Diagnostic Information	No	As there is no state externalization, there is also no analysis on periodic diagnostic information.
COM:05	RAIL:04	Component Restart Service	Yes	The TAS platform provides recovery for CEs. However, state externalization of middleware is insufficient for CE recovery and the applications are often too complex for periodic externalization - there is need for iterative recovery concepts for applications.

### 3.2.1.6. External Memory Management Services

Genesys defines three optional memory management services. The following table contains their mapping to the TAS platform.

Genesys Optional Services (External Memory Management Services) → TAS Platform				
Req.	Feature	Service Name	Important for Railway domain (yes/no)	Rationale for Importance / Implementation in Tas Platform
-	-	Access Control of Memory Partitions	No	An external memory access mechanism is not used in the TAS platform - the platform solely uses local memories and strictly message based communication between the CEs (no shared memory).
-	-	Stable Storage	No	An external memory access mechanism is not used in the TAS platform - the platform solely uses local memories and strictly message based communication between the CEs (no shared memory).
-	-	Secure Storage	No	An external memory access mechanism is not used in the TAS platform - the platform solely uses local memories and strictly message based communication between the CEs (no shared memory).

### 3.2.1.7. Security Services

Genesys defines six optional security services. The following table contains their mapping to the TAS platform.

Genesys Optional Services (Security Services) → TAS Platform				
Req.	Feature	Service Name	Important for Railway domain (yes/no)	Rationale for Importance / Implementation in Tas Platform
-	-	Secure Key Management	No	The TAS platform does not offer any inherent security services, as it operates on a closed network.
-	-	Encryption and Decryption	Yes	Encryption of communication messages can be done via external mechanisms.

-	-	Random Number Generation	No	The TAS platform does not offer any inherent security services.
-	-	Service Authentication	No	The TAS platform does not offer any inherent security services.
-	-	Secure Boot Service	No	The TAS platform does not offer any inherent security services.
-	-	Service Access Control	No	The TAS platform does not offer any inherent security services.

### 3.2.1.8. Resource Management Services

Genesys defines four optional resource management services. The following table contains their mapping to the TAS platform.

Genesys Optional Services (Resource Management Services) → TAS Platform				
Req.	Feature	Service Name	Important for Railway domain (yes/no)	Rationale for Importance / Implementation in Tas Platform
ECR:02	-	Local Resource Management	Yes	Local resource management at CE level is done by the operating system (time/scheduling management, memory management). QoS management is also of growing importance.
-	-	Global Resource Management	No	There is no additional global resource management to the basic configuration services.
-	-	Device Level Resource Management	No	There is no additional device resource management to the basic configuration services.
-	-	Configuration and Reconfiguration	No	The TAS platform does not allow node reconfiguration at runtime. The addition of a CN has to be handled by the application and the communication interface.

### 3.2.1.9. Gateway Services

Genesys defines five optional gateway services. The following table contains their mapping to the TAS platform.

<b>Genesys Optional Services (Gateway Services) → TAS Platform</b>				
<b>Req.</b>	<b>Feature</b>	<b>Service Name</b>	<b>Important for Railway domain (yes/no)</b>	<b>Rationale for Importance / Implementation in Tas Platform</b>
-	-	Wireless Connection	Yes	The TAS platform uses gateway services at CE level, i.e. in an Moon system a gateway is implemented N times. This enables the use of redundant communication channels and makes a dedicated gateway interface unnecessary. So far, wireless connections are not directly supported by TAS, but that is an important issue.
-	RAIL:06	Internet Connection	Yes	Currently, the TAS control system is a closed network, so there is no necessity for an internet gateway. However, this feature is also an important issue for TAS.
-	RAIL:06	Legacy Integration	Yes	Communication between different node versions of the TAS platform is supported through the inter-node communication protocol OCS. This is only for inter CE communication, not for intra CE.
-	RAIL:13	Fault-tolerant Clock Synchronization	Yes	The TAS platform uses a fault tolerant algorithm to generate the common time base on node level.
-	-	Process Input Output	Yes	In the railway domain many sensors and actuators (e.g. axle counters, signals, switches) must be managed by the system through process I/O.

### 3.2.1.10. Mobility Services

Genesys defines four optional mobility services. The following table contains their mapping to the TAS platform.

<b>Genesys Optional Services (Mobility Services) → TAS Platform</b>				
<b>Req.</b>	<b>Feature</b>	<b>Service Name</b>	<b>Important for Railway domain (yes/no)</b>	<b>Rationale for Importance / Implementation in Tas Platform</b>
-	-	Component / Service Detection	No	Dynamic configuration changes are not allowed in the TAS platform, therefore no component/service detection is necessary.

-	-	Connectivity Management	No	Devices are stationary for the TAS platform concept.
-	-	Mobile Device Controlled Mobility	No	The TAS platform does not require mobility.
-	-	Infrastructure Controlled Mobility	No	The TAS platform does not require mobility.

### 3.2.1.11. Generic Middleware Services

Genesys defines four optional generic middleware services. The following table contains their mapping to the TAS platform.

Genesys Optional Services (Generic Middleware Services) → TAS Platform				
Req.	Feature	Service Name	Important for Railway domain (yes/no)	Rationale for Importance / Implementation in Tas Platform
COM:10	RAIL:08 RAIL:09	Voting Service	Yes	Voting is done at CE level on all CEs, i.e. in an MooN system a message can be voted N times.
ECR:02	-	Event Recognition and Handling	Yes	Events like incoming messages or timer interrupts are handled by the operating system's scheduler on CE level.
-	-	High-level Protocol Implementation	Yes	The TAS platform uses a high-level one channel safe (OCS) protocol for communication between CNs.
-	-	Receiver Controlled Streaming	No	Data streaming is not important for the TAS platform.

## 4. Requirements

In the following requirements for the establishment of error containment regions and external communication between Computing Nodes shall be stated.

### 4.1. Error containment regions

Building error containment regions with respect to physical faults by component replication is costly and not energy effective. It could be shown that an N-version programming approach for application software not only addresses software specification and implementation faults but also renders error containment regions with respect to physical faults on a single hardware component [GRUB01]. However, this approach is costly for complex software applications and might fail to provide the required error detection coverage for software that implements trivial functionality. Application functionality is difficult to describe diversely if the implemented functionality is simple such as for communication protocols.

To overcome these limitations of N-version programming, automated diversity seems to be promising. Given the testability of the application software to exclude implementation faults, the automated diversity approach derives two diverse execution binaries from a single source base. The diverse binaries have to exercise the system resources (hardware and operating system) such that physical faults manifest differently in both software channels.

TRSS-AT:REQ:ECR:01	
Synopsis	Automated diversity
Description	Provide methods and tools for automated software diversity to provide resilience with respect to physical faults on a single channel
Status	Active
Priority	Mandatory

For safety-critical applications the high degree of system complexity and the reuse of existing components bear a considerable risk for unknown design faults in the hardware and the operating system. For modern processors and their accompanying hardware components (memory interface, peripheral interfaces) errata sheets are being published and updated periodically revealing great numbers of design/implementation faults. From a safety point of view it has to be concluded that each hardware component bears undetected hardware bugs at the time of field deployment of the safety-critical product. The same is true for off-the-shelf and/or open source operating systems. With respect to this class of faults neither spatial separation nor temporal separation as well as simple replication of components is a mitigation strategy.

Design faults require special attention when integrating COTS hard- and software components. To cope with unknown hardware design faults a straight forward approach is to employ diverse hardware implementations for system components.

To assess the impact of unknown operating system faults in non-software diverse systems, an investigation of robustness services has to be carried out by research on pseudo-random behavior of complex operating systems in the context of loosely coupled fault-tolerant architectures. The aim is to clarify the impact of loose coupling on the system's ability to tolerate intricate faults (e.g., races) in the underlying operating system. Results gained for a homogeneous hardware setup have to be compared to a setup using diverse hardware, running two separate operating system instantiations for diverse processor architectures.

<b>TRSS-AT:REQ:ECR:02</b>	
Synopsis	Integration of COTS operating systems
Description	Provide analysis of operating system behavior in loosely coupled replicated systems for homogeneous components and hardware diverse components
Status	Active
Priority	Mandatory

## 4.2. Communication

This section states the requirements on the communication interfaces in the railway domain. As no paradigm change is planned for the inter-replica communication interface and the local interfaces of TAS platform, this section concentrates on the external interface for inter CN communication. In the railway signaling domain two communication scenarios can be identified: Local area networks connecting subsystems within the indoor part of an interlocking system and wide area networks for the connection of trackside system components such as signals, point machines and axle counters. The requirements stated below cover these two communication aspects.

<b>TRSS-AT:REQ:COM:01</b>	
Synopsis	Wide Area Network transmission speed
Description	Communication shall be possible over Wide Area Networks with transmission speed of 64kBps or higher.
Status	Active
Priority	Mandatory

<b>TRSS-AT:REQ:COM:02</b>	
Synopsis	Wide Area Network transmission distance
Description	Communication shall be possible over Wide Area Networks with a geographical extension in the range of 1km to 100 km repeaters and modems shall be possible
Locality	WAN
Status	Active
Priority	Mandatory

<b>TRSS-AT:REQ:COM:03</b>	
Synopsis	Wide Area Network real time capabilities
Description	Real-time capabilities for wide area networks shall be in the range of 200ms
Locality	WAN
Status	Active
Priority	Mandatory

<b>TRSS-AT:REQ:COM:04</b>	
Synopsis	Local Area Network real time capabilities
Description	Real-time capabilities for local area networks shall be in the range of 10ms
Locality	LAN
Status	Active
Priority	Mandatory

<b>TRSS-AT:REQ:COM:05</b>	
Synopsis	Reconfiguration in case of faults
Description	The network shall be reconfigurable in the case of faults
Locality	WAN and LAN
Status	Active
Priority	Mandatory

<b>TRSS-AT:REQ:COM:06</b>	
Synopsis	Virtual channels
Description	The communication system shall support multiple virtual channels over the same physical link to the same or different logical peers.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

<b>TRSS-AT:REQ:COM:07</b>	
Synopsis	Quality of service
Description	The communication system shall support configurable quality of service in the channels.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

<b>TRSS-AT:REQ:COM:08</b>	
Synopsis	Network topology
Description	The communication system shall support different network topologies, at least ring and star.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

<b>TRSS-AT:REQ:COM:09</b>	
Synopsis	Physical redundancy
Description	The communication system shall support redundant physical communication links.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

<b>TRSS-AT:REQ:COM:10</b>	
Synopsis	Redundancy configuration
Description	The communication system shall be configurable as hot-hot, warm or cold standby
Locality	WAN and LAN
Status	Active
Priority	Mandatory

TRSS-AT:REQ:COM:11	
Synopsis	Transmission Errors
Description	The communication shall be robust to cover transient transmission errors by using retransmission or other corrective means.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

TRSS-AT:REQ:COM:12	
Synopsis	Packet oriented transmission
Description	The communication system shall work in a packet oriented way.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

TRSS-AT:REQ:COM:13	
Synopsis	Standard compliance
Description	The communication system shall conform to EN 50159-1 and EN 50159-2.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

TRSS-AT:REQ:COM:14	
Synopsis	Serving redundant applications
Description	The communication system shall be usable from redundant applications running on 1 or more CPUs and care for selecting and checking proper data.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

TRSS-AT:REQ:COM:15	
Synopsis	Determinism in the Time Domain
Description	The communication system be deterministic in the time domain
Locality	WAN and LAN
Status	Active
Priority	Optional

## 5. Features of domain specific platform solution

Implementing the Software-based TTEthernet protocol into diverse computing elements of a computing node on TAS Platform will increase the deterministic behavior of the communication system with the advantages of a Time Triggered Communication system. This topic will provide an overview of TTEthernet features followed by the mapping of this features to the requirements defined in chapter 4.

### 5.1. TTEthernet Introduction

As depicted in Figure 2, the time-triggered services (TT Services) can be viewed parallel to the common OSI layers: a communication controller (Hardware-based) or a software layer in combination with a COTS Ethernet controller (Software-based) that implements the TT Services is able to synchronize its local clock with the local clocks of other communication controllers and switches in the system. The communication controller can then send messages at off-line planned points in this synchronized global time. These messages are said to be time-triggered and it is the task of the off-line planning tool to guarantee that the time-triggered message schedule is free of conflict. By conflict-free we mean that it will never be the case that two time-triggered messages compete for transmission and, hence, no dynamic arbitration actions for the communication medium (for time-triggered messages) are required. TTEthernet supports communication among applications with different real-time and safety requirements over a single physical network. Therefore, three different traffic classes are provided (see Figure 2): Time-Triggered Traffic (TT), Rate-Constrained (RC) Traffic, and Best-Effort Traffic (BE). If required, the corresponding traffic class of a message can be identified based on a message's Ethernet Destination address.

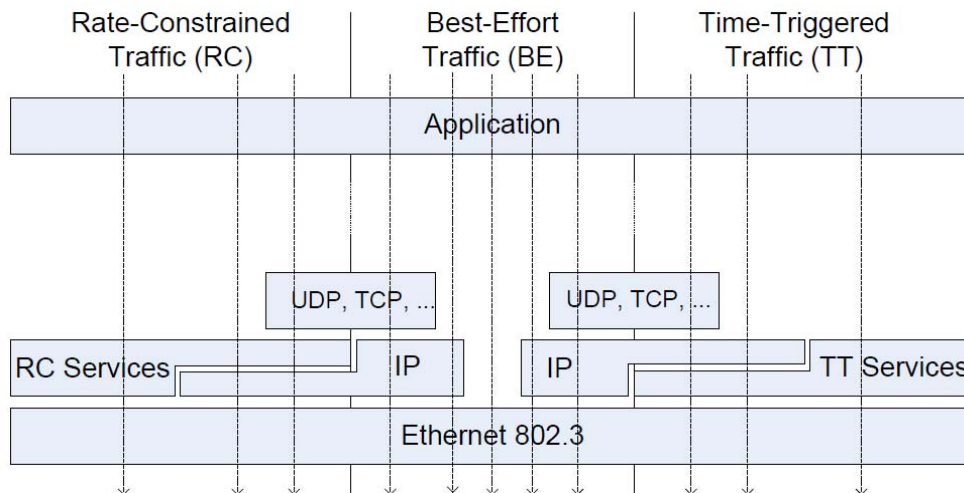


Figure 2: Interaction of standards

As depicted in Figure 2, messages from higher layer protocols, like IP or UDP, can easily be "made" time-triggered without modifications of the messages' contents itself. This is because the TTEthernet protocol overhead is transmitted in dedicated messages, called Protocol Control Frames, which are used to establish system-wide synchronization upon those components that need to be synchronized. In short, TTEthernet is only concerned with "when" a data message is sent, rather than with specific contents within a data message.

TTEthernet is a transparent clock synchronization protocol, which means that it is able to co-exist with other traffic, potentially legacy traffic, on the same physical communication network.

## 5.2. Software-Based TTEthernet

The Software-based TTEthernet specifies a special implementation of TTEthernet which was created to make use of the time-triggered communication benefit implemented in software without the fault tolerant features of the hardware-based solution enabling also high throughputs for much lower costs. If the use case requires fault tolerance mechanisms then they have to be implemented in the application on top of the TTEthernet API Library.

Currently the software-based TTEthernet implementation is based on 100MBit/s Ethernet Controller and the 100MBit/s TTEthernet switch offering a low cost solution of a TTEthernet implementation.

Figure 3 shows the layered structure of the software-based TTEthernet implementation on a host without an operating system including the TTEthernet protocol core layer embedded between the hardware layer and the API library.

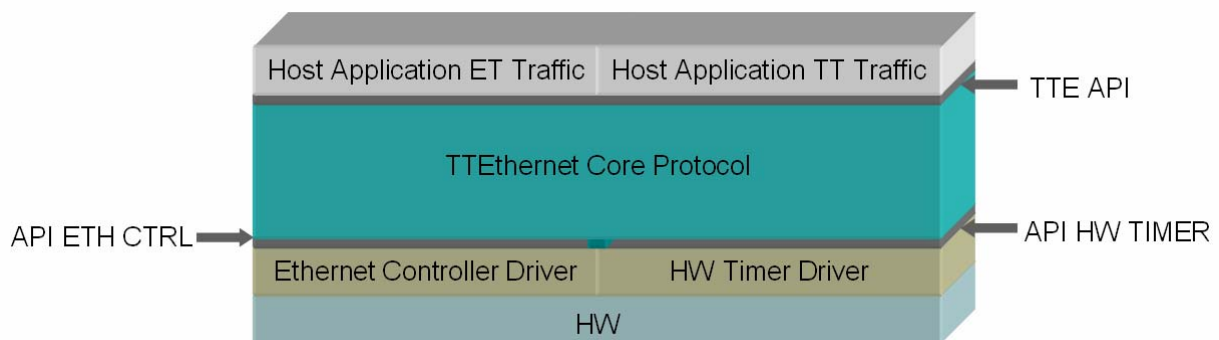


Figure 3: Software-Based TTEthernet detailed architecture

Setting up an application with software-based TTEthernet on a platform with an operating system a driver has to be established on top of the TTEthernet core protocol to get access to the core functions (Figure 4).

- On top of this TTEthernet OS Driver layer the API library will provide functions for
- TTEthernet protocol management (e.g. initialize-, configure the end-system)
  - Functions for sending and receiving data
    - Time-triggered messages
    - Rate-constraint messages
    - Best-effort messages
  - Auxiliary functions (flush buffers, get controller state)

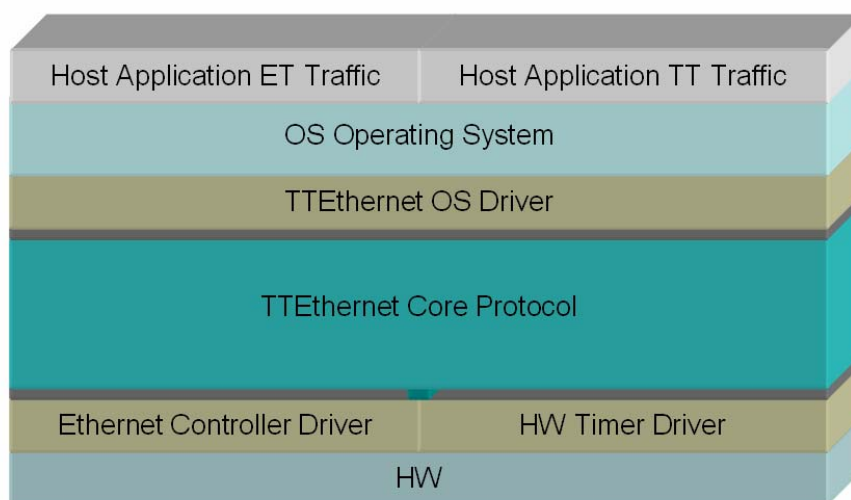


Figure 4: Software-Based TTEthernet with Operating System Driver

### 5.3. Scheduling Mechanisms for TTEthernet

Requisites to ensure a dependable system play a major role in railway applications. The deterministic characteristics of TTEthernet thus qualify its use in this domain. Basically, TTEthernet classifies communication traffic in three distinct classes:

- **time-triggered (TT) messages** are sent over the network at predefined times. They take precedence over all other traffic type and thus ensure temporal determinism for message delivery. Messages with hard real-time requirements are time-triggered;
- **rate-constrained (RC) messages** are granted with network bandwidth guarantees and with limited delays and temporal deviations. Typically, multimedia applications are classified in this category.
- **event-triggered (ET) messages** are treated as classical Ethernet messages. Based on best-effort principle they are transmitted during the idle periods after TT and RC messages have been scheduled. Thus it is not possible to guarantee whether or when these messages can be transmitted.

Even though the definitions of these classes of messages are clear, the appropriate classification of individual messages is not always straightforward. Some messages correspond to inherent strictly periodic behavior suitable directly for the TT approach. Sporadic messages, i.e., with unknown arrival times, but minimum time intervals between consecutive instances, can be classified either as TT or RC messages. If these messages are classified as time-triggered with a period corresponding to their worst case period, the minimum, timely correctness is guaranteed. If their actual arrival patterns at run time, however, are not strictly periodic, reserved slots will be underutilized. On the other hand, if they are classified as rate-constrained messages no time slots will be wasted but the jitter introduced on the message delivery caused by this classification might not be acceptable by the application.

Besides temporal correctness, other constraints must be considered for scheduling the message delivery. For instance, message transmission must be scheduled so that the buffers on the network switches do not overflow. The recently published research [ISO99] and [ISO00] will be used as start point to solve the issues mentioned above. Even though they handle scheduling of tasks on CPU, the proposed methods will be investigated so that they can be applied for network scheduling.

GENESYS offers currently services for reading and writing periodic, sporadic and real-time stream messages. However, the scheduling policy used to schedule these messages has not been addressed yet. Therefore, new approaches for both off-line and on-line scheduling mechanisms of messages in TTEthernet are now under investigation. The application in different domains will be studied.

## 5.4. Mapping of TTEthernet features to the TAS Platform requirements

The software-based TTEthernet features and the mapping to the TAS Platform requirements can be found in the following in this section.

TTT:RAIL:01	
Synopsis	Portability
Description	The Software-based TTEthernet protocol is portable to any general-purpose computer supplying a COTS NIC and a hardware timer. Therefore it has been designed in a layered-structure consisting of a core layer including the TTEthernet core functionality, a hardware layer including the driver for the COTS NIC and the hardware timer, and an API-library.
Locality	CE
Status	Active
Priority	Mandatory

TTT:RAIL:02	
Synopsis	Wide Area Network real time capabilities
Description	The Software-based TTEthernet uses the physical layer (PHY) of the COTS Ethernet-hardware from the host target. For long distances a PHY with fiber-optical ports can be used to reach small latency and jitter. To maintain the signal quality over very long distances repeaters can be used to improve the signal quality.
Locality	WAN
Status	Active
Priority	Mandatory

TTT:RAIL:03	
Synopsis	Local Area Network real time capabilities
Description	<p>The application of transparent clock mechanism allows to precisely re-establish temporal order of synchronization messages.</p> <p>In a first step the worst case delay is calculated off-line. In a second step, each synchronization message is delayed for "worst case delay minus dynamic delay" upon reception of the synchronization message, where the dynamic delay is the delay added to the synchronization message, as the synchronization message flows through the communication channel. The point in time at "worst case delay minus dynamic delay" after the Reception Point in Time will be called the Permanence Point in Time. The novel concept of Permanence Point in Time allows precisely calculating the earliest point in time when a message becomes permanent allowing the exact calculation of the latency. Therefore the jitter of a transmission is not cumulative. Typical latency values are smaller than 200µs per hop and jitters smaller than 100µs.</p>
Locality	LAN
Status	Active
Priority	Mandatory

TTT:RAIL:04	
Synopsis	Reconfiguration in case of faults
Description	The API-Library offers functions to configure the TTEthernet core protocol layer. In case of faults reconfiguration of this core protocol layer is one possible recovery process.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

TTT:RAIL:05	
Synopsis	Virtual link schema
Description	Critical (Time-Triggered or Rate Constrained) Traffic in TTEthernet is based the virtual link (VL) schema defined in ARINC 664 part 7. A Virtual Link is a path between sender and receiver(s) (1:n relation) with a unique VL identifier (VL ID). Switches are configured to know about the VL definitions (up to 4096 VL IDs per switch).
Locality	WAN and LAN
Status	Active
Priority	Mandatory

TTT:RAIL:06	
Synopsis	Quality of service
Description	Traffic classes differ by the quality-of-service (QoS) they provide. In TTEthernet, we distinguish between three traffic classes: <ul style="list-style-type: none"> <li>• time-triggered (TT),</li> <li>• rate-constrained (RC), and</li> <li>• best-effort (BE) traffic.</li> </ul>
Locality	WAN and LAN
Status	Active
Priority	Mandatory

TTT:RAIL:07	
Synopsis	Network topology
<i>Continued on next page →</i>	

← *Continued from previous page*

Description	TTEthernet end systems are connected to switches via bi-directional communication links. An end system will communicate with a second end system or a group of end systems via sending a message to the switch, which will then relay the message to the receiving end system or end systems. Also, switches can be connected to each other via bi-directional communication links. In this case we call the resulting architecture a multi-hop architecture and the links between any two switches the multi-hop link.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

<b>TTT:RAIL:08</b>	
Synopsis	Physical redundancy
Description	Porting Software-based TTEthernet to different targets to make use of a hardware diverse platform can be achieved with small effort during its layered design.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

<b>TTT:RAIL:09</b>	
Synopsis	Redundancy configuration
Description	Executing two or more instances of the Software-based TTEthernet at the same time by one host makes it possible to handle active redundant data or to run the multiples as hot-, warm- or cold-standby communicating components.
<i>Continued on next page →</i>	

← <i>Continued from previous page</i>	
Locality	WAN and LAN
Status	Active
Priority	Mandatory

<b>TTT:RAIL:10</b>	
Synopsis	Transmission Errors
Description	All TT messages are sent over the network at predefined times and take precedence over all other traffic classes. Furthermore all messages are checked for correctness (CRC) at the receiver. In case, where an end system decides not to use its assign timed slot, for example if there is no new data to be sent, the switch recognizes the inactivity of the sender and frees the bandwidth for the other traffic classes. The handling of transient transmission errors within the time-triggered traffic class has to be coped during design time of the network by assigning shorter periods to the messages to get multiple instances at the receiver. Transmitting messages more often as required will reduce the bandwidth of the communication channel but also dispense the overhead during requesting retransmissions.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

<b>TTT:RAIL:11</b>	
Synopsis	Packet oriented transmission
Description	Implementing the protocol in the low level IEEE 802 layers performs services transparently within the Data Link layer, using all IEEE 802.3 services without modification and not modifying IEEE 802.2 services. Therefore, the communication frames in TTEthernet follow the IEEE 802.3 frame format.
<i>Continued on next page</i> →	

← Continued from previous page

Locality	WAN and LAN
Status	Active
Priority	Mandatory

TTT:RAIL:12	
Synopsis	Determinism in the Time Domain
Description	Time-Triggered (TT) messages are used for applications with tight determinism requirements. All TT messages are sent over the network at predefined times and take precedence over all other traffic classes.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

TTT:RAIL:13	
Synopsis	Global Time Service
Description	IEEE 1588 [IEEE08] IEEE 1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE, 2008. specifies a synchronization protocol for Ethernet. The global time base of TTEthernet can be leveraged to synchronize also native IEEE 1588 synchronization clients. For this purpose, additional functionality can be realized on top of a TTEthernet device that generates IEEE 1588 clock synchronization frames. TTEthernet provides means to compensate for delays through the TTEthernet network. Outside the TTEthernet network, in a native IEEE 1588 network, the clock synchronization messages can be handled as native IEEE 1588 clock synchronization messages.
Locality	WAN and LAN
Status	Active
Priority	Mandatory

## 6. References

1. [INDEXYS\_D1.1] A. Balogh, Gy. Csertán, A. Ökrös, Z. Balogh, P. Bokor, G. Fohler, R. Coelho, "Report on initial project alignment according to final GENESYS results", Edition 1.0, 2009-06-30
2. [EN\_50128] "Railway Applications: Software for Railway Control and Protection Systems", CENELEC EN 50128, March 2001
3. [EN\_50129] "Railway Applications: Safety Related Electronic Systems for Signaling", CENELEC ENV 50129, 2003
4. [EN\_50159\_1] "Railway applications: Communication, signaling and processing systems, CENELEC EN 50159 Part 1: Safety-related communication in closed transmission systems", 2001
5. [EN\_50159\_2] "Railway applications: Communication, signaling and processing systems, CENELEC EN 50159 Part 2: Safety-related communication in open transmission systems", 2001
6. [POLE96] S. Poledna: "Fault Tolerant Real-Time Systems: The Problem of Replica Determinism", Kluwer Academic Publishers, Boston, 1996.
7. [GRUB01] Thomas Gruber, Walter Böhm, "Elektronische Stellwerke – Fehlertolerante Systeme im Technologiewandel", IT'S T.I.M.E, 01/2001 p 61
8. [IEEE08] IEEE 1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE, 2008.
9. [ISO99] D. Isovich and G. Fohler, Online Handling of Hard Aperiodic Tasks in Time Triggered Systems, In Proceedings of the 11th Euromicro Conference on Real-Time Systems, 1999.
10. [ISO00] D. Isovich and G. Fohler, Efficient Scheduling of Sporadic, Aperiodic and Periodic Tasks with Complex Constraints, In Proceedings of the 21st IEEE RTSS, 2000.
11. [GENESYS] R. Obermaisser and H. Kopetz (Eds.), GENESYS: A Candidate for an ARTEMIS Cross-Domain Reference Architecture for Embedded Systems: Südwestdeutsche Verlag für Hochschulschriften, 2009.