

DOCUMENT

Document / Deliverable Name: **Analysis and prioritization of exploitable cross-domain services with focus on automotive**

Document / Deliverable Nr. **D 2.1**

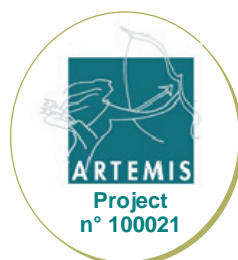
Version: **draft** **1.0**
 final

Document Type: **confidential**
 public

Responsible: **R. Obermaisser, TUVI**

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 Signalling Solutions GmbH	TRSS-AT	Austria
08	Technical University of Darmstadt	TUDA	Germany
09	Technical University of Kaiserslautern	UNIKL	Germany
10	Vienna University of Technology	TUVI	Austria

Author(s)

Name	Company
R. Obermaisser	TUVI
R. Kammerer	TUVI
M. Schlager	TTT
E. Schmidt	TTT

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

Revision chart and history log

Version	Date	Reason
0.1	2009-08-09	Initial Version
0.2	2009-10-07	Added FlexRay Multi Switch Feature Descriptions
0.3	2009-10-30	Added Mapping to GENESYS Reference Architecture Template
1.0	2009-12-30	Correction of typos, requirements moved into annex

Table of Contents

1. INTRODUCTION.....	6
1.1 FlexRay Multi-Switch	6
1.2 CAN Router.....	7
2. FEATURES OF DOMAIN-SPECIFIC PLATFORM SOLUTION.....	9
3. MAPPING OF FEATURES TO GENESYS REFERENCE ARCHITECTURE TEMPLATE.....	17
4.1 Core Architectural Services	18
4.1.1 Basic Configuration Services	18
4.1.2 Basic Execution Control Services	18
4.1.3 Basic Time Services	18
4.1.4 Basic Communication Services.....	19
4.2 Optional Architectural Services.....	19
4.2.1 Diagnostic Services	19
4.2.2 Resource Management Services	19
4.2.3 Legacy Integration	20
4. REFERENCES	21
5. ANNEX 1 – Requirements for CAN Router	
6. ANNEX 2 – Requirements for FlexRay Multi-Switch	

List of Figures

Figure 1: FlexRay Cluster with Multi-Switch.....	6
Figure 2: Example FlexRay cluster schedule with multi-switch.....	7
Figure 3: CAN Router	8
Figure 4: Example for Mapping from Bus Topology (left) to Star Topology (right).....	8
Figure 5: Services of the Reference Architecture Template	17

1. Introduction

FlexRay [1] and Controller Area Network (CAN) [2] are two communication protocols with high significance in the automotive domain. This document analyses requirements and features for architectural services based on these communication protocols. The architectural services shall be provided in a FlexRay multi-switch and a CAN router, which serve as the core of automotive communication infrastructures. In addition, the requirements and features are analysed in the context of the GENESYS architecture [3]. We evaluate whether the GENESYS cross-domain services are suitable to satisfy the defined requirements and features.

1.1 FlexRay Multi-Switch

The FlexRay “multi-router” is a cut-through switch for FlexRay networks. It switches a number of FlexRay branches according to a pre-defined static schedule. The communication elements are forwarded with minimal delay and are not stored and forwarded at a later point in time. Therefore, we decided to change the name from “multi-router” to “**multi-switch**”.

A FlexRay multi-switch is a device which is physically similar to the FlexRay Active Star device, but in contrast to the Active Star, it implements a selective switching of the communication paths according to a configured switching schedule. The multi-switch is able to provide additional functionality of complex data traffic paths and also isolation of branches (cf. next figure).

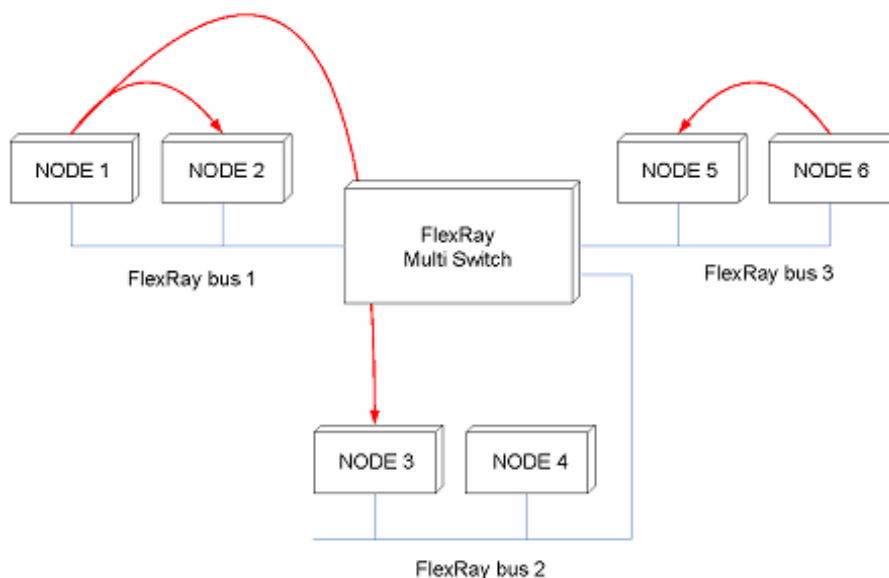


Figure 1: FlexRay Cluster with Multi-Switch

In this example FlexRay busses 1 and 2 are connected together during specific static slots via the FlexRay multi-switch into one single cluster whereas the bus 3 is isolated and can perform its own private communication. In the above example a two times higher bandwidth is achieved in a single static slot compared to a bus or active star system. An example for such a communication schedule for such a system is shown in the next figure.

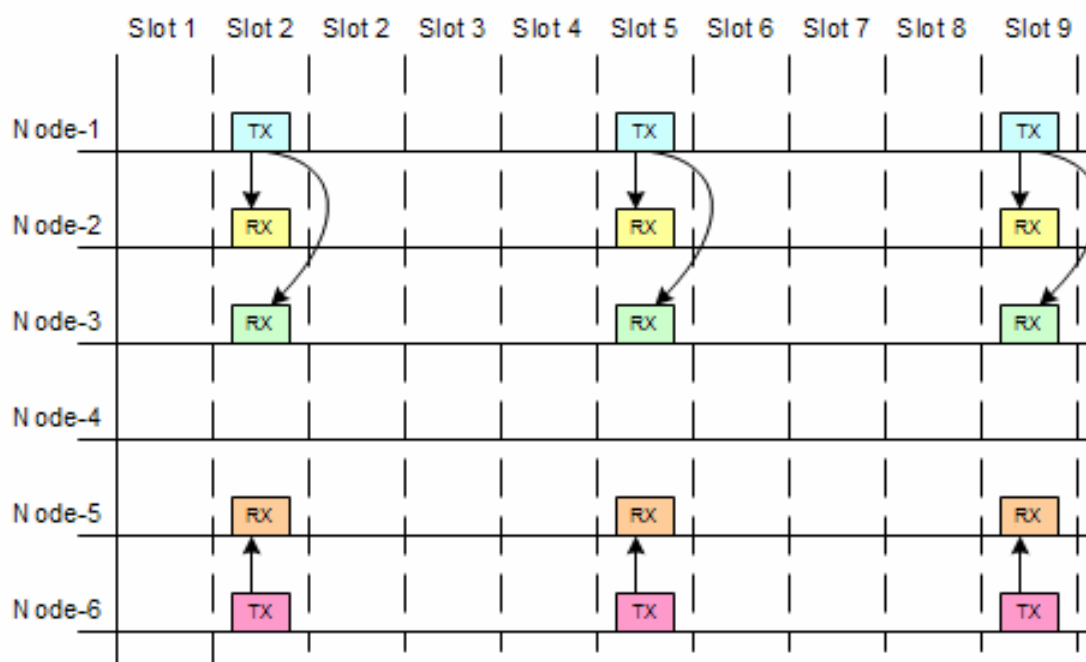


Figure 2: Example FlexRay cluster schedule with multi-switch

1.2 CAN Router

CAN is the most widely used protocol in today's automotive electronic systems. Present day cars contain multiple CAN buses deployed for different domains such as comfort or power-train subsystems. Properties of CAN that have led to its success include its simplicity, high flexibility, efficiency and low cost. Adversely, CAN exhibits limitations with respect to robustness, scalability, and ease of system integration. An example of a hazard to robustness is the failure assumption for nodes. A single node can cause a global failure of the communication system by constantly sending high priority messages. From the point of view of scalability, bandwidth and wire length limitations, which are caused by CAN's arbitration mechanism, result in tight bounds for the scale of CAN-based systems. Also, during system integration, the temporal properties of exchanged messages change. The latencies of a message increase as nodes are integrated that send messages with higher priorities. In case of real-time applications, this effect can cause failures if a certain number of nodes is exceeded.

This report proposes requirements for a CAN router, which overcomes the limitations of CAN while keeping the desirable properties of CAN that have led to its success. When using the CAN router, a star topology is used instead of the bus topology of conventional CAN networks. The CAN router tackles scalability by improving performance, possible wire length and supporting multiple namespaces. The CAN router improves performance by multicasting messages only to those nodes that require a message. Through the star topology, individual physical lines are shorter compared to a bus topology, enabling a higher overall wire length. In addition, the CAN router improves robustness by blocking messages from faulty CAN nodes, thereby preventing the corruption of the entire system. System integration is eased via strict control on the interaction patterns between nodes.

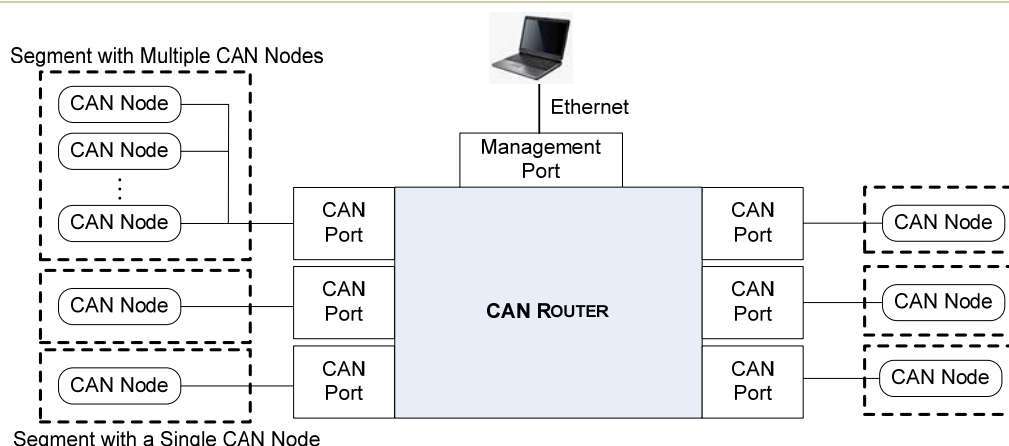


Figure 3: CAN Router

Figure 3 depicts a CAN system using the CAN router. The CAN router possesses a set of CAN ports, each of which provides the connection to the CAN bus of a corresponding CAN segment. A CAN segment consists of one or more CAN nodes. Small segments (ideally with only a single node) maximize the fault isolation and performance benefits through the CAN router. Controlled by its configuration, the CAN router redirects messages between the CAN segments. In addition to the CAN ports, the CAN router possesses a management port. The management port serves for changing the configuration of the CAN router and for the retrieval of diagnostic information.

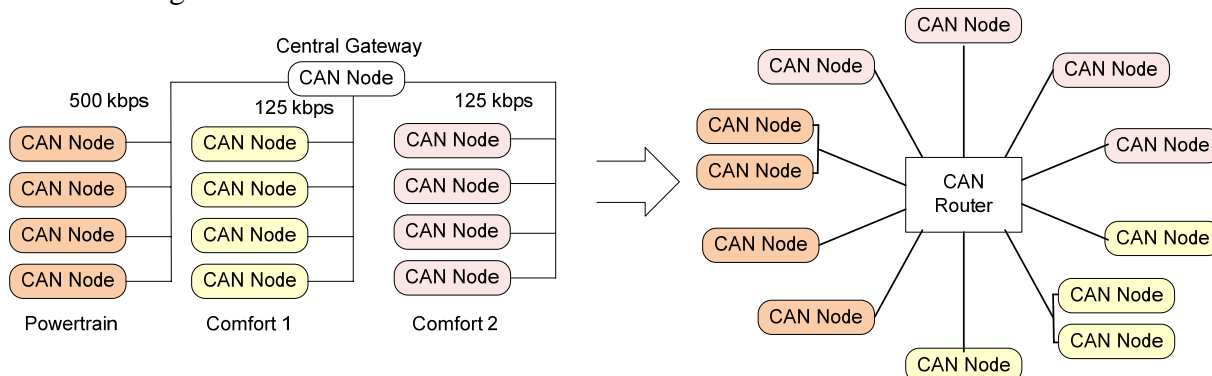


Figure 4: Example for Mapping from Bus Topology (left) to Star Topology (right)

An example for the mapping from today's bus-based systems to the proposed star topology is depicted in Figure 4. CAN nodes of different domains (e.g., powertrain, comfort in the example) can be connected to the CAN router.

2. Features of Domain-Specific Platform Solution

ID	Auto-CAN-1
Description	Enforcement of minimum message interarrival times: In order to achieve fault isolation in the temporal domain, the CAN router ensures that message transmissions comply with specified minimum message interarrival times. In a properly configured system, the CAN router limits the effect of messages sent by one CAN segment onto the temporal properties of messages sent by other CAN segments (e.g., latencies, variability of latencies). The configuration of the CAN router encompasses for each CAN segment and each redirected message a specification of the minimum message interarrival time.
Importance	High
Rationale	This feature supports the requirements of composable integration and fault isolation.
Implementation Hints	

ID	Auto-CAN-2
Description	Enforcing of valid message identifiers: The CAN router ensures that a CAN node from one CAN segment cannot masquerade as a CAN node from another CAN segment. Therefore, the configuration of the CAN router includes for each CAN segment a pool of permitted message identifiers. A message with an identifier that is not contained in this pool is blocked by the CAN router.
Importance	High
Rationale	This feature supports the requirements of composable integration and fault isolation.
Implementation Hints	

ID	Auto-CAN-3
Description	Electrical compatibility and protocol compatibility: The interaction between a CAN node and the CAN router will adhere to the CAN Specification ISO 11898/11519. The CAN router will provide a differential 2-wire interface for connecting a Shielded Twisted Pair (STP), Un-shielded Twisted Pair (UTP), or Ribbon cable. The Bit Encoding used is: Non Return to Zero (NRZ) encoding (with bit-stuffing) for data communication on the differential two wire bus. The CAN bus interface will use an asynchronous transmission scheme controlled by start and stop bits at the beginning and end of each message. Information is passed from transmitters to receivers in a data frame, which is composed of an arbitration field, control field, data field, CRC field and ACK field. The frame begins with a start-of-frame field

	and ends with an end-of-frame field. The data field may be from 0 to 8 bytes. A frame check sequence is computed with a Cyclic Redundancy Code (CRC). Both the standard (11 bit identifier) and extended CAN formats (29 bit identifier) will be supported.
Importance	High
Rationale	This feature supports the requirements of “Legacy CAN-interface support”.
Implementation Hints	It is planned to deploy commercial-off-the-shelf CAN transceivers and CAN controllers in the router.

ID	Auto- CAN-4
Description	Maximum latency: The CAN router will introduce a maximum overhead in the redirection for CAN messages of 2 ms. (note: this overhead does not include the delays imposed by competing higher priority messages)
Importance	High
Rationale	The realization of real-time applications with the CAN router requires bounds on the overhead for the redirection of messages.
Implementation Hints	The design and implementation of the router will occur in such a way that the processing delays do not incur an overhead of more than 2 ms.

ID	Auto-CAN-5
Description	Multicasting: For each message received from a CAN segment, the router determines to which other segments the message needs to be redirected. Two special cases of multicasting are broadcast and point-to-point communication topologies. Broadcasting is the topology of a conventional CAN network, delivering a message to all CAN segments and thus all CAN nodes. In case of point-to-point communication, a message is redirected to a single CAN segment only.
Importance	High
Rationale	Multicasting results in a more efficient use of the communication bandwidth than broadcasting. This feature supports the requirement “exceed limits of CAN”.
Implementation Hints	The configuration of the CAN router incorporates routing tables in order to permit the message multicasting.

ID	Auto-CAN-6
Description	Overall bandwidth of more than 1 Mbps: The CAN router supports a star topology and multicasting. Although the bandwidth of an individual wire is limited to 1 Mbps, the overall (system-level) bandwidth can be higher than 1 Mbps.
Importance	High
Rationale	This feature supports the requirement “exceed limits of CAN”.
Implementation	

Hints	
--------------	--

ID	Auto-CAN-7
Description	Wire length of more than 40m: The CAN router supports a star topology and multicasting. Although the length of an individual wire is limited to 40 m, the overall (system-level) wire length can be longer than 40 m.
Importance	High
Rationale	This feature supports the requirement “exceed limits of CAN”.
Implementation Hints	

ID	Auto-CAN-8
Description	Name translation: The CAN router will support the conversion of message identifiers. For each CAN port, source message identifiers with mappings to destination message identifiers for the other CAN ports can be specified.
Importance	High
Rationale	This feature supports the requirement “exceed limits of CAN”.
Implementation Hints	The configuration of the CAN router will contain tables that define the mapping of identifiers between different CAN nodes.

ID	Auto-CAN-9
Description	Priority ordering of messages: During arbitration in CAN, each transmitting node monitors the bus state and compares the received bits with the transmitted bit. If a dominant bit is received when a recessive bit is transmitted then the node stops transmitting (i.e., it lost arbitration). As a consequence, high-priority messages are transmitted on the CAN bus before competing messages with lower priority. The CAN router will establish the priority ordering for CAN messages. In case multiple messages shall be redirected to a CAN segment, the CAN router will do so in the order of their priorities. A message stored within the CAN router is overwritten in case a newer message with the same identifier arrives.
Importance	High
Rationale	This feature is essential for many CAN-based applications. In addition, it supports the requirement of “Legacy CAN-interface support”.
Implementation Hints	It is planned to deploy a priority queue at each CAN port.

ID	Auto-CAN-10
Description	Logging of diagnostic information: The CAN router collects information about violations of the minimum message interarrival times and invalid message identifiers.
Importance	Medium
Rationale	The diagnostic information is an important basis for maintenance deci-

	sions, e.g., whether an ECU needs to be replaced.
Implementation Hints	

ID	Auto-CAN-11
Description	Retrieval of diagnostic information via a management port: Using a management port, diagnostic information can be retrieved. The retrieval of diagnostic information does not disrupt the routing of CAN messages and can also occur during normal operation.
Importance	Medium
Rationale	
Implementation Hints	It is planned to provide an Ethernet interface in the CAN router to allow the retrieval of diagnostic information.

ID	Auto-CAN-12
Description	Reconfiguration of CAN router via management port: Using the management port, the configuration of the CAN router can be modified at run-time. Examples of possible reconfiguration actions are: <ul style="list-style-type: none"> • modification of a message's minimum and maximum message interarrival time • addition and removal of messages • enabling/disabling of ports • modification of multicast patterns (i.e., different sets of receiving nodes) • modification of name translation
Importance	Medium
Rationale	Dynamic reconfiguration is important for mass customization and dynamically changing application contexts.
Implementation Hints	

ID	Auto-CAN-13
Description	Consistent switch over to new configuration: A new configuration can be programmed, which becomes active at a specified point time at all CAN ports. In particular, no inconsistent or intermediate configurations are used in the redirection of messages through the CAN router while the re-configuration is in progress.
Importance	Medium
Rationale	Inconsistent or intermediate configurations could result in application-level failures.
Implementation Hints	It is planned to store two configurations: the active configuration and a shadow configuration. The shadow configuration can be reprogrammed, while the active configuration is used to control the behavior of the CAN router. After the reprogramming is completed, the active and shadow con-

	figurations can be switched.
--	------------------------------

ID	Auto-CAN-14
Description	<i>Sleep mode:</i> The CAN router will support a sleep mode, in which all messages except for special wake-up messages are blocked. The arrival of a wake-up message shall terminate the sleep mode. Sleep is entered upon a corresponding command from the management port.
Importance	Medium
Rationale	Electronic Control Units (ECUs) should have a very small sleep mode current during key-off. The parking time may be as long as 3 to 6 months.
Implementation Hints	It is planned to build upon the dynamic reconfiguration capabilities in order to implement the sleep mode.

ID	Auto-CAN-15
Description	<i>Synchronization message:</i> For each segment, the CAN router can be configured to send a periodic message with the global time. The synchronization message can be individually enabled or disabled for each segment. Also, the identifier and period of the synchronization can be configured.
Importance	Medium
Rationale	The synchronization message supports the establishment of a global time base.
Implementation Hints	In order to achieve a global time based with a high precision, the synchronization message should be assigned the highest CAN priority.

ID	Auto-MS-16
Description	<i>Device Interfaces:</i> The FlexRay multi-switch will be equipped with the following interfaces: <ul style="list-style-type: none"> • FlexRay branches with FlexRay transceivers. • Communication controller interface (local FX communication controller) • Host interface (SPI)
Importance	High
Rationale	This feature supports the requirements “included in an ECU” and “accompanied by a host micro controller” [Requ-Auto-MS-2, Requ-Auto-MS-3].
Implementation Hints	

ID	Auto-MS-17
Description	<i>Electrical compatibility and protocol compatibility:</i> The multi-switch will be fully compatible to a FlexRay active star device according to FlexRay Electrical Physical Layer Specification 2.1B.

Importance	High
Rationale	This feature supports the requirement of FlexRay EPL 2.1B compatibility [Requ-Auto-MS-1].
Implementation Hints	It is planned to deploy commercial-off-the-shelf FlexRay Multi Switch devices which are compatible to recent active star devices.

ID	Auto-MS-18
Description	Multicasting and parallel message transmissions: For each message received from a FlexRay branch, the router determines to which other branches the message needs to be redirected. There will be the possibility to transmit two or more messages at the same time, given that these messages do not use the same set of branches.
Importance	High
Rationale	Multicasting results in a more efficient use of the communication bandwidth than broadcasting and further allows multiple messages to be transmitted at the same time (in parallel). This feature supports the requirement “multiple input/output branches” [Requ-Auto-MS-7].
Implementation Hints	The configuration of the multi-switch incorporates schedule switching configuration in order to permit message multicasting.

ID	Auto-MS-19
Description	Bit reshaping: The multi-switch supports bit reshaping which means that an incoming message is regenerated by the multi-switch while it is transmitted on outgoing branches.
Importance	Medium
Rationale	This feature supports the requirement “bit reshaping” [Requ-Auto-MS-14].
Implementation Hints	The bit reshaping needs to occur with minimal propagation delay.

ID	Auto-MS-20
Description	Sleep mode: The multi-switch will support a sleep mode, in which all messages except for special wake-up messages are blocked. The arrival of a wake-up message shall terminate the sleep mode. Sleep is entered upon a corresponding command from a host command.
Importance	High
Rationale	Electronic Control Units (ECUs) should have a very small sleep mode current during key-off. The parking time may be as long as 3 to 6 months. This feature supports the requirement “sleep mode” [Requ-Auto-MS-4].
Implementation Hints	

ID	Auto-MS-21
-----------	------------

Description	<i>Cascading:</i> The FlexRay multi-switch will be equipped with an interconnection interface for cascading several multi-switch devices.
Importance	Medium
Rationale	This feature supports the requirement “cascaded multi-switch devices” [Requ-Auto-MS-13].
Implementation Hints	

ID	Auto-MS-22
Description	<i>Wakeup pattern forwarding:</i> The multi-switch will support a fast transition from Sleep Mode to Operational Mode in order to forward a sufficient number of wakeup patterns out of a burst of incoming wakeup patterns which wake up the multi-switch itself.
Importance	High
Rationale	That way the multi-switch supports fast wakeup of a complete FlexRay cluster. This feature supports the requirements “FlexRay Wakeup Patterns” and “wake-up procedure” [Requ-Auto-MS-5, Requ-Auto-MS-10].
Implementation Hints	

ID	Auto-MS-23
Description	<i>FlexRay Synchronization:</i> The multi-switch passively synchronizes to the FlexRay communication.
Importance	High
Rationale	The multi-switch needs to know the points in time when FlexRay slots start / end. This feature supports the requirement “FlexRay synchronization” [Requ-Auto-MS-6].
Implementation Hints	

ID	Auto-MS-24
Description	<i>Central Bus Guardian:</i> During synchronous operation, the multi-switch supports bus guardian functionality in the sense that it enforces a pre-defined switching schedule.
Importance	High
Rationale	The multi-switch prevents error propagation from faulty nodes by suppressing (faulty) messages which are sent in the wrong slot. This feature supports the requirements “Central Bus Guardian” and “confine broadcast of erroneous sources” [Requ-Auto-MS-8, Requ-Auto-MS-9].
Implementation Hints	

ID	Auto-MS-25
Description	Cycle Multiplexing: The multi-switch supports a switching configuration which allows individual data paths for each slot. Thereby it is possible to assign different switching configurations for each FlexRay cycle.
Importance	Medium
Rationale	Cycle multiplexing is necessary to optimize the bandwidth utilization due to the fact that the communication schedule can be different for each FlexRay cycle. This feature supports the requirement “cycle multiplexing” [Requ-Auto-MS-11].
Implementation Hints	It must be avoided that cycle multiplexing leads to an intolerable increase of the necessary switching configuration data.

ID	Auto-MS-26
Description	Store and Forward: The multi-switch may support Store and Forward functionality for a limited number of messages (this is an optional feature and it will be discussed during the implementation phase if it makes sense to support it).
Importance	Low
Rationale	Store and forward provides additional flexibility to build FlexRay clusters without the need to reconfigure all ECUs after a schedule change. This feature supports the requirement “store and forward” [Requ-Auto-MS-12].
Implementation Hints	

ID	Auto-MS-27
Description	Dynamic Segment: The multi-switch shall support the possibility of configuring broadcast groups which exchange messages within the dynamic segment. Communication in the dynamic segment is non-deterministic. In the dynamic segment, the multi-switch performs similar to an active star device but with the difference that several broadcast groups can exchange data in parallel.
Importance	Medium
Rationale	This feature supports the requirement “dynamic segment” [Requ-Auto-MS-15].
Implementation Hints	

3. Mapping of Features to GENESYS Reference Architecture Template

The GENESYS reference architecture template [3] provides architectural services (cf. Figure 5) as a baseline for the development of applications. GENESYS distinguishes between core services and optional services. The core services are mandatory in every GENESYS-based system. They provide a stable baseline for the realization of applications and higher architectural services. The optional architectural services are optional in the sense that they are not required in every instantiation of the architecture. If needed, developers can pick them out of the GENESYS reference architecture template, which includes a set of existing service specifications for the different levels of integration. The set of domain-independent optional services is open, i.e., new optional services can be added if the need arises.

In the following the identified requirements and features (cf. Sections 2 and 3) are mapped to the GENESYS reference architecture template. This section starts with an analysis of the relationship between the core architectural services and the requirements/features. Thereafter, optional services are identified that facilitate the establishment of those requirements and features that are not yet covered through the core services.

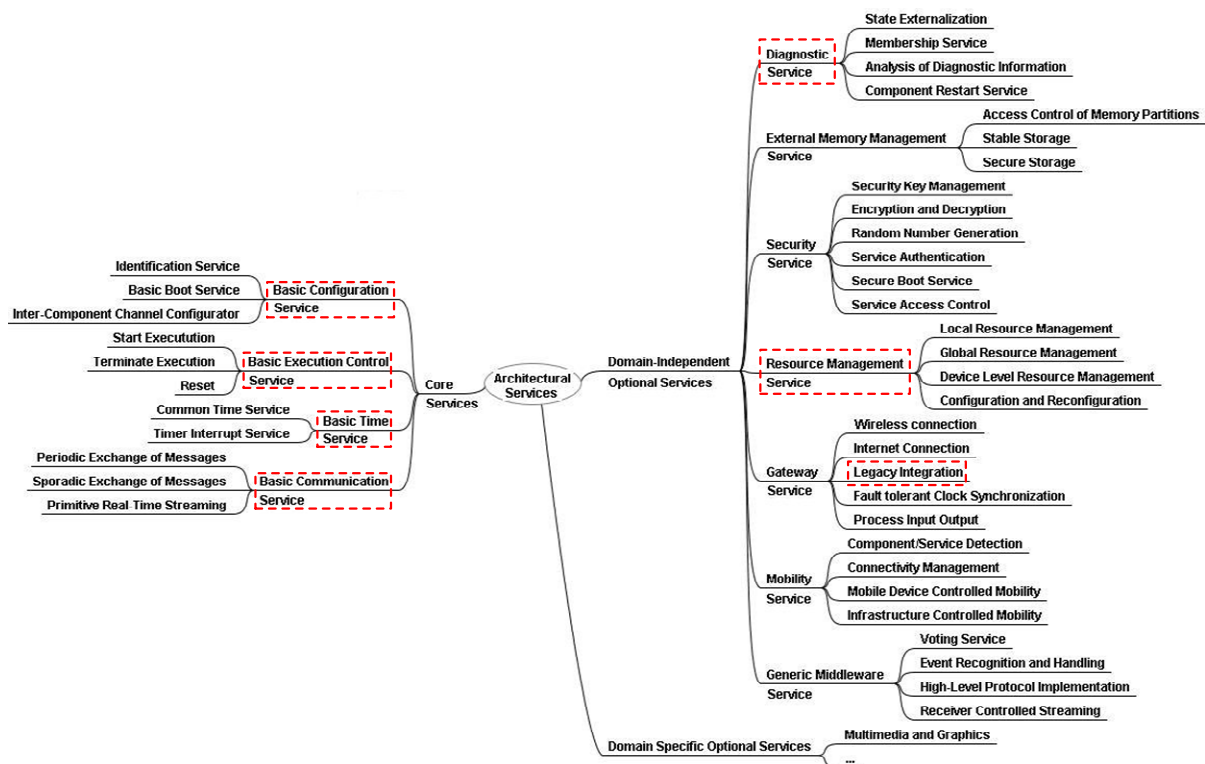


Figure 5: Services of the Reference Architecture Template

4.1 Core Architectural Services

The reference architecture template offers four core services: basic configuration, basic execution control, basic time and basic communication. These services map to several of the automotive requirements and features, thus motivating the deployment of all core services in the FlexRay multi-switch and the CAN router.

4.1.1 Basic Configuration Services

Basic configuration services are necessary to satisfy the introduced automotive requirements and provide the identified features. The reconfiguration of the CAN router occurs via a management port. As described by Requirement Req-Auto-CAN-6 and Feature AUTO-CAN-12, examples of use cases for the management port include the addition/removal of messages, the enabling/disabling of ports and the modification of the name translation.

In particular, the capabilities requested by the requirements and features include the inter-component channel configurator and the identification service. Information about valid message identifiers (CAN interconnect, Requirement Req-Auto-CAN-5) and permitted sending slots (FlexRay multi-switch, Feature Auto-MS-24) are used for fault isolation and the prevention of masquerading failures. In addition, the ability for a consistent switch over to new configurations is defined by the introduced features.

4.1.2 Basic Execution Control Services

The presented requirements and features support the basic execution control. The FlexRay multi-switch features Auto-MS-20 (“Sleep mode”) and Auto-MS-22 (“Wakeup pattern forwarding”) are concerned with basic execution control. For the CAN interconnect, feature Auto-CAN-14 (“Sleep mode”) is concerned with execution control of CAN-based end systems.

4.1.3 Basic Time Services

The basic time services are covered by the FlexRay-based systems. FlexRay-based automotive applications use a global time base to coordinate the communication activities. The support for a global time base is reflected in Feature Auto-MS-23 “FlexRay Synchronization”. The timer interrupt service is not within the scope of the described features and requirements, because existing endsystems are to be used in conjunction with the FlexRay multi-switch.

The CAN router supports the establishment of a common time base through Requirement Req-Auto-CAN-7 (clock synchronization support) and Feature Auto-CAN-15 (synchronization message). The CAN router can act as a time master that synchronizes the attached CAN segments. The synchronization messages are configurable (i.e., identifier and synchronization period). The priority of the synchronization message determines the jitter of the synchronization messages and thus the precision of the global time base.

4.1.4 Basic Communication Services

The FlexRay Multi-Switch and the CAN router support the basic communication modes of GENESYS. The FlexRay system performs the periodic exchange of messages through the time-triggered communication patterns in the static segments (Auto-MS-18, Auto-MS-23). The sporadic exchange of messages occurs using the dynamic segment (Feature Auto-MS-27).

A CAN-based system is inherently based on sporadic event-triggered communication. Periodic communication is supported as a particular use case of sporadic communication. Due to the enforcement of minimum message interarrival times in the CAN router (Feature Auto-CAN-1), temporal guarantees (i.e., bounded latency) can be given for both periodic and sporadic messages.

4.2 Optional Architectural Services

The following optional architectural services of the reference architecture template facilitate the establishment of the identified requirements and features.

4.2.1 Diagnostic Services

The design of the diagnostic services of the GENESYS architecture differentiates between active and passive diagnosis. In passive diagnosis, the detected errors are only logged (e.g., for later maintenance actions). Passive diagnosis cannot interfere with the application behaviour and is thus not safety-related even if the monitored application is safety-related.

Requirement Req-Auto-CAN-5 defines passive diagnostic capabilities of the CAN router. The CAN router shall record timing and value message failures. In the time domain, violations of the minimum message interarrival time shall be recorded. In the value domain, information about naming incoherence will be collected.

4.2.2 Resource Management Services

Resource management refers to the ability of an embedded system to dynamically adapt to the context. Applications can have multiple application quality levels, which are selected dynamically based on the user preferences and platform resource availability.

The CAN router supports dynamic resource management and provides a management port to change the configuration of the CAN system.

The reconfiguration capabilities of the CAN router map to optional services for resource management in the GENESYS reference architecture template. These reconfiguration capabilities extend the capabilities of the basic configuration services (i.e., the inter-component channel configuration). For example, the name translation in the CAN router can be config-

ured (cf. Feature AUTO-CAN-12). Also, the permitted temporal behaviour of the CAN router can be set in order to adapt the error detection to different operational scenarios.

4.2.3 Legacy Integration

Legacy systems can represent major investments and a complete redevelopment of these systems is often unacceptable due to cost and time constraints. For this purpose both the CAN router and the FlexRay Multi-Switch provide communication interfaces that are fully compatible to existing CAN-based and FlexRay-based systems. Requirement Req-Auto-CAN-3 (“Legacy CAN-interface support”) captures this requirement for the CAN router. This requirement is reflected in Feature Auto-CAN-3 (“Electrical compatibility and protocol compatibility”). For the FlexRay multi-switch, electrical compatibility and protocol compatibility is defined in Feature Auto-MS-17.

4. References

- [1] FlexRay Consortium, "FlexRay Communications System Protocol Specification Version 2.1," 2005.
- [2] Robert Bosch GmbH, "CAN Specification, Version 2.0," 1991.
- [3] R. Obermaisser and H. Kopetz (Eds.), *GENESYS: A Candidate for an ARTEMIS Cross-Domain Reference Architecture for Embedded Systems*: Südwestdeutsche Verlag für Hochschulschriften, 2009.

Annex 1 – Requirements for CAN Router

ID	Req-Auto-CAN-1
Description	<p><i>Facilitate Composable Integration:</i> The CAN router needs to provide a framework that supports the smooth integration and reuse of independently developed components in order to increase the level of abstraction in the design process.</p> <p>A key requirement in order to ensure composability is the support for non-interfering interactions at the communication system. If there exist two disjoint subgroups of cooperating components that share a common communication medium, then the communication activities within one subgroup may not interfere with the communication activities within the other subgroup. If this principle is not satisfied, then the integration within one component-subgroup depends on the proper behavior of the other component-subgroups.</p>
Importance	High
Rationale	Composability significantly reduces integration efforts and facilitates the management of complexity.
Implementation Hints	<p>The CAN router will <i>selectively redirect messages</i>, i.e., only specific segments receive a particular message. Hence, disjoint groups of CAN nodes, which do not exchange messages, cannot influence the message exchanges of each other.</p> <p>In addition, the enforcement of <i>minimum message interarrival times</i> will bound the effect that interacting segments can have on the timing of each other's message exchanges.</p>

ID	Req-Auto-CAN-2
Description	<p><i>Exceed limits of CAN (bandwidth, cable length, name space):</i> The CAN router shall exceed the limitations of conventional bus-based CAN systems with respect to performance, cable length and message identifiers. The arbitration mechanism of the CAN protocol limits bandwidth to 1 Mbps (at a cable length of 40m), because CSMA/CA requires bits to stabilize on the network. Therefore the bit length must be at least as long as the propagation delay of the network. The CAN router enables the transition from a bus topology to a star topology. In contrast to a bus, a star topology encompasses multiple CAN wires. Although the bandwidth of an individual wire is limited to 1 Mbps, the overall (system-level) bandwidth can be higher than 1 Mbps.</p>

	<p>The namespace of CAN is determined by the CAN identifier bits (i.e., 11 bits in standard CAN format, 29 bits in extended format). In the absence of global coordination in the use of identifiers, naming incoherence can occur. A naming incoherence is the use of the same identifier in different CAN messages or the assignment of different identifiers to the same CAN message by sets of CAN nodes. An example for the latter case would be a node that sends a CAN message with a particular identifier, although some receivers expect to receive the message with a different identifier. The CAN router shall resolve naming incoherence by converting between message identifiers.</p>
Importance	High
Rationale	
Implementation Hints	<p>The CAN router shall multicast messages only to the nodes that require a message. Most messages need not be broadcast to all nodes, but only to a subset of the nodes. Through the presence of multiple wires at the overall system-level instead of a single one, the CAN router also permits longer networks than possible in a bus which is limited to 40m at 1 Mbps.</p> <p>For the conversion of message identifiers, the configuration of the CAN router will contain tables that define the mapping of identifiers between different CAN nodes.</p>

ID	Req-Auto-CAN-3
Description	<p>Legacy CAN-interface support: Many existing CAN-based components are available. The development of these components represents major investments, which must not be compromised through the migration to a CAN communication system based on the CAN router. Therefore, the CAN router must be fully compatible to a CAN bus. Firstly, the ports of the CAN router must be electrically compatible to standard CAN. Secondly, the interaction with CAN nodes at each port must occur according to the CAN protocol. For example, this implies the use of arbitration through CSMA/CA even if only a single node is connected to a port. Also, the CAN router has to acknowledge received messages.</p>
Importance	High
Rationale	Without full compatibility to legacy systems, high cost would result due to the redevelopment of already existing CAN-based components.
Implementation Hints	It is planned to construct the CAN router using standard CAN transceivers and commercial-off-the-shelf CAN controllers to ensure full

	compatibility to the CAN standard.
--	------------------------------------

ID	Req-Auto-CAN-4
Description	<p><i>Fault isolation:</i> A bus-based CAN system possesses several limitations concerning fault isolation. For example, a faulty node can disrupt the communication of all other nodes in case it continuously sends high priority messages. Another example is the susceptibility to masquerading faults. A faulty node can send messages with wrong identifiers, thereby overwriting messages of correct nodes at the receivers.</p> <p>The CAN router shall support fault isolation in the temporal domain by enforcing predefined temporal specifications (e.g., minimum message interarrival times). In addition, the CAN router shall support fault isolation in the value domain by blocking messages with invalid message identifiers.</p>
Importance	High
Rationale	Fault isolation significantly improves the robustness of the system by ensuring that nodes affected by design faults, as well as transient or permanent hardware faults cannot corrupt the rest of the system.
Implementation Hints	The configuration of the CAN router will include a priori knowledge about the permitted behavior of CAN nodes in the time and value domain. This a priori knowledge will be used to block messages from faulty CAN nodes.

ID	Req-Auto-CAN-5
Description	<p><i>Diagnostic Support:</i> The CAN router shall record violations of the specification in the value and time domain. In the time domain, violations of minimum and maximum message interarrival times shall be recorded. In the value domain, information about naming incoherence will be collected.</p> <p>Using a management port of the CAN router, it shall be possible to retrieve this diagnostic information.</p>
Importance	High

Rationale	Though the breakdown logs of today's ECUs inform the service technician about detected errors within the system, they do not assist the technician adequately in the identification process. Thus, fully functional units are replaced, or even worse, faulty ECUs remain unchanged in the system.
Implementation Hints	The configuration of the CAN router, which is deployed for enabling fault isolation, will be used to monitor the behavior of CAN nodes. In case messages are blocked by the fault isolation mechanisms, diagnostic records will be stored for later retrieval by a maintenance engineer.

ID	Req-Auto-CAN-6
Description	Reconfiguration support: The CAN router shall support the dynamic modification of its configuration in order to add/remove CAN nodes and modify the specified communication patterns of the CAN nodes (e.g., used identifiers, minimum message interarrival times).
Importance	Medium
Rationale	CAN-based systems need to adapt to changing application contexts without requiring the redevelopment of the CAN infrastructure.
Implementation Hints	Using the management port, the CAN router will accept reconfiguration commands, which control the modification of the CAN router's configuration.

ID	Req-Auto-CAN-7
Description	Clock synchronization support: The CAN router shall support the establishment of a global time base.
Importance	Medium
Rationale	A global time base is important for coordinating distributed activities and for establishing a relationship between time stamps assigned at different end systems.
Implementation Hints	

Annex 2 – Requirements for FlexRay Multi-Switch

ID	Requ-Auto-MS-1
Description	A switch shall be fully compatible with an active star as defined in <ul style="list-style-type: none"> ○ [FlexRay EPL 2.1B timing, electrical values etc.], and in ○ the OEM specification
Rationale	compatibility to existing FlexRay networks when switching not needed, cheaper device possible
Importance	High

ID	Requ-Auto-MS-2
Description	A switch shall always be included in an ECU
Importance	High

ID	Requ-Auto-MS-3
Description	A switch shall always be accompanied by a host μ C
Rationale	This might prevent the switch from containing flash memory, which would be the third technology on the same chip (or even die), as the host could configure the switch every boot time / every time it boots)
Importance	High

ID	Requ-Auto-MS-4
Description	A switch shall support sleep mode (also via host command)
Importance	High

ID	Requ-Auto-MS-5
Description	A switch shall distribute FlexRay Wakeup Patterns very fast after Wakeup of the device itself
Importance	High

ID	Requ-Auto-MS-6
Description	A switch shall synchronize to the FlexRay clock, to meet the FlexRay timing requirements
Importance	High

ID	Requ-Auto-MS-7
Description	A switch shall switch multiple input branches (0-*) to multiple output branches (0-*)
Importance	High

ID	Requ-Auto-MS-8
Description	A switch shall implement some features of the central bus guardian
Rationale	it may also be used for connecting x-by-wire subsystems with the remaining vehicle.
Importance	High

ID	Requ-Auto-MS-9
Description	A switch shall confine the broadcast of erroneous sources to the slots allocated to their branches.
Importance	High

ID	Requ-Auto-MS-10
Description	A switch shall implement the wake-up procedure as specified for the Active Star.
Importance	High