ETSI TS 103 641 V1.1.1 (2019-03)

TECHNICAL SPECIFICATION

Reconfigurable Radio Systems (RRS);
Radio Equipment (RE) reconfiguration requirements

Reference
DTS/RRS-0217

Keywords
CRS, mobile, SDR

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Reconfigurable Radio Systems (RRS).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The scope of the present document is to define the high level system requirements for reconfigurable Radio Equipment enabling the provision of Radio Applications. The work is based on the Use Cases defined in ETSI TR 103 062 [i.1], ETSI TR 102 944 [i.2], ETSI TR 103 585 [i.3] and ETSI EN 302 969 [i.4].

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 103 062: "Reconfigurable Radio Systems (RRS); Use Cases and Scenarios for Software Defined Radio (SDR) Reference Architecture for Mobile Device".

[i.2] ETSI TR 102 944: "Reconfigurable Radio Systems (RRS); Use Cases for Baseband Interfaces for Unified Radio Applications of Mobile Device".

[i.3] ETSI TR 103 585: "Reconfigurable Radio Systems (RRS); Radio Equipment Reconfiguration Use Cases".

[i.4] ETSI EN 302 969: "Reconfigurable Radio Systems (RRS); Radio Reconfiguration related Requirements for Mobile Devices".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

distributed computations: model in which components located on networked computers communicate and coordinate their actions by passing messages interacting with each other in order to achieve a common goal

Functional Block (FB): function needed for real-time implementation of Radio Application(s)

NOTE 1: A functional block includes not only the modem functions in Layer1 (L1), Layer2 (L2), and Layer 3 (L3) but also all the control functions that should be processed in real-time for implementing given Radio Application(s).

NOTE 2: Functional blocks are categorized into *standard functional blocks* and *user defined functional blocks*. In more details:

1) *Standard functional blocks* can be shared by many Radio Applications. For example, Forward Error Correction (FEC), Fast Fourier Transform (FFT)/Inverse Fast Fourier Transform (IFFT), (de)interleaver, Turbo coding, Viterbi coding, Multiple Input Multiple Output (MIMO), Beamforming, etc are the typical category of standard functional block.

2) *User defined functional blocks* include those functional blocks that are dependent upon a specific Radio Application. They are used to support special function(s) required in a specific Radio Application or to support a special algorithm used for performance improvement. In addition, a user defined functional block can be used as a baseband controller functional block which controls the functional blocks operating in baseband processor in real-time and to control some context information processed in real-time.

NOTE 3: Each functional block has its unique name, Input, Output and properties.

network coding: technique in which transmitted data is encoded and decoded to improve network performance

Radio Application (RA): software which enforces the generation of the transmit RF signals or the decoding of the receive RF signals

NOTE 1: The Software is executed on a particular radio platform or an RVM as part of the radio platform.

NOTE 2: Radio applications might have different forms of representation. They are represented as:

- source codes including Radio Library calls of Radio Library native implementation and Radio HAL calls;
- Intermediate Representations (IRs) including Radio Library calls of Radio Library native implementation and radio HAL calls;
- Executable codes for a particular radio platform.

radio library: library of Standard Functional Blocks (SFB) that is provided by a platform vendor in a form of platform-specific executable code

NOTE 1: SFBs implement reference codes of functions which are typical for radio signal processing. They are not atomic and their source codes are typed and visible for Radio Application developers.

NOTE 2: An SFB is implemented through a Radio Hardware Abstraction Layer (HAL) when the SFB is implemented on dedicated HW accelerators. Radio HAL is part of ROS.

Radio Virtual Machine (RVM): abstract machine supporting reactive and concurrent executions

NOTE: A Radio Virtual Machine may be implemented as a controlled execution environment which allows the selection of a trade-off between flexibility of base band code development and required (re-)certification efforts.

reconfigurable radio equipment: Radio Equipment with radio communication capabilities providing support for radio reconfiguration

NOTE: Reconfigurable Radio Equipment includes Smartphones, Feature Phones, Tablets, Laptops, Connected Vehicle communication platform, Network platform, IoT device, etc.

resources: Hardware Resources that a Radio Application needs in active state

NOTE 1: Resources are provided by the reconfigurable Radio Equipment (RE), to be used by the Radio Applications when they are active. Radio Applications provide their Resource needs (e.g. using operational states) so that the multiradio computer may judge whether these Resources are available, in order to ensure non-conflicting operation with other Radio Applications. Resources may or may not be shared in the reconfigurable RE.

NOTE 2: Resources may include processors, accelerators, memory, Radio Frequency circuitry, etc.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASIC	Application Specific Integrated Circuit
BER	Bit Error Rate
CAT	CATegory
CR	Cognitive Radio
FB	Functional Block
FEC	Forward Error Correction
FFT	Fast Fourier Transform
HAL	Hardware Abstraction Layer
IoT	Internet of Things
IR	Intermediate Representation
LTE	Long Term Evolution
MAC	Media Access Control
MIMO	Multi-Input Multi-Output
MU-MIMO	Multi User-Multi-Input Multi-Output
PER	Packet Error Rate
PMI	Precoding Matrix Indicator
RA	Radio Application
RAT	Radio Access Technology
RE	Radio Equipment
RERC	Radio Equipment Reconfiguration Class
RF	Radio Frequency
RI	Rank Indicator
ROS	Radio Operating System
RRS	Reconfigurable Radio Systems
RSSI	Received Signal Strength Indication
RVM	Radio Virtual Machine
Rx	Receive
SDR	Software Defined Radio
SFB	Standard Functional Block
SINR	Signal to Interference-plus-Noise Ratio
SU-MIMO	Single User-Multi-Input Multi-Output
Tx	Transmit
UDFB	User Defined Functional Block
WiFi	Wireless Fidelity

4 Requirement Organization and Methodology

4.0 General

This clause is containing the description of how the requirements are organized and the related format.

4.1 Requirement Organization

As shown in Figure 1, all requirements described in the present document belong to one single category (the functional requirements category). Requirements are, in turn, organized into groups.

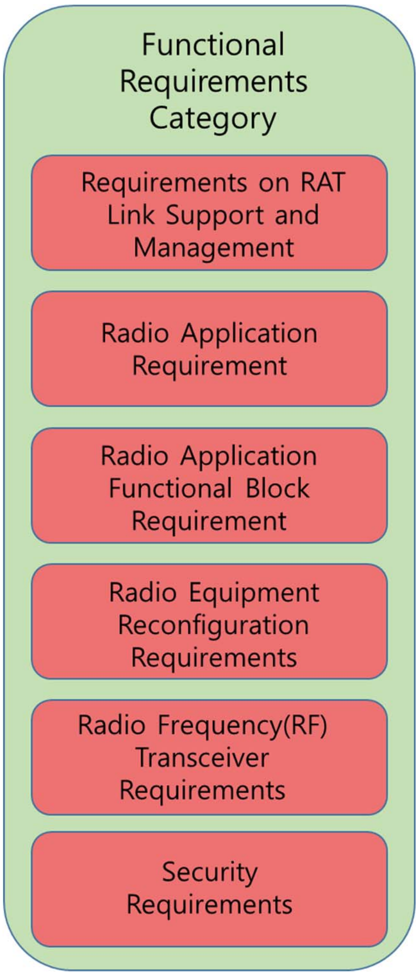

Figure 1: Overall requirements structure

4.2 Requirement Format

A letter code system is defined which makes a unique identification of each requirement R-<CAT>-<GROUP>-<XX>. Each requirement is constructed as follows:

- R-: Standard requirement prefix.
- <CAT>

Code	Category
FUNC	Functional aspects

- <GROUP>: Requirement group identifier. A letter code will be used for this identifier. The three first letters will give the identifier of the group.
- <XX>: Requirement identifier within requirement group; range 01 = > 99.

EXAMPLE: R-FUNC-QOS-01.

4.3 Requirement Formulation

A requirement is formulated in such a way that it is uniquely defined. It is built as follows:

Title: <Title Description>

- Description: the description of a requirement will be formulated using the terms as described in the clause "Modal verbs terminology" above.

5 Working assumptions

5.1 Assumptions

5.1.1 Radio Equipment Reconfiguration Classes

As it is expected that the reconfiguration capabilities of a Radio Equipment will evolve over time, Radio Equipment Reconfiguration Classes (RERC) are introduced. As shown in Figure 2, 7 different classes of reconfigurable RE are introduced (RERC-0 corresponds to a non-reconfigurable device).

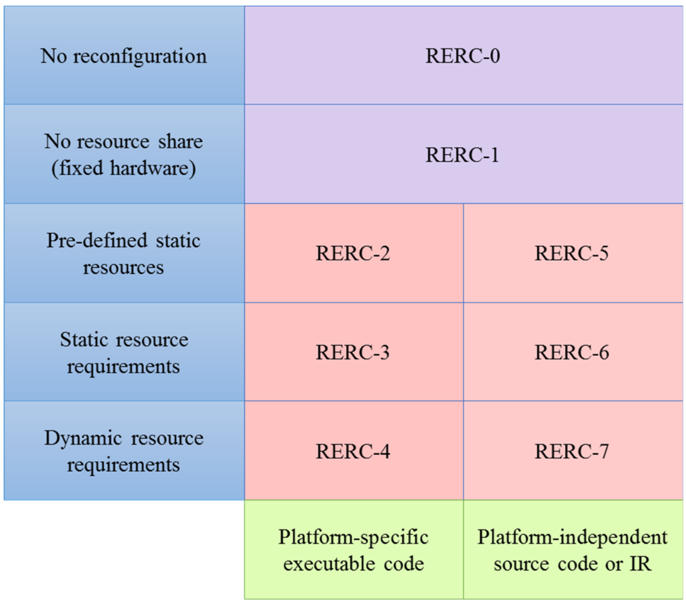

Figure 2: Definition of RERCs according to reconfiguration capabilities

A reconfigurable RE belongs to a defined class according to the reconfiguration capabilities, which are determined by the type of Resource requirements and the form of the Radio Application Package. Reconfigurable RE classes are defined as follows (see also Figure 2):

1) RERC-0: No RE reconfiguration is possible.

RERC-0 represents legacy radio implementations and do not allow for RE reconfiguration (except for bug fixing and release-updates through firmware updates) or exploitation of Cognitive Radio (CR) features. RERC-0 represents legacy radio implementations and does not allow for RE reconfiguration.

2) RERC-1: Radio Applications use different fixed Resources.

In this scenario, at least some of the radios are implemented with non-software defined radio (SDR) technology, e.g. with dedicated Application Specific Integrated Circuits (ASICs), and are Resource-wise independent of each other. Simple CR functionality may be supported through radio parameter management to the extent which the radio implementations allow. RERC-1 implements multiple Radio Applications with fixed Resources allocation and no Resource sharing.

The rule for Resource allocation for multiple applications $\{A_1, A_2, \ldots, A_N\}$ can be formulated as follows: $A_i \rightarrow R_i$, $\forall i \in \{1, ..., N\}$, where R_i denotes Resources allocated for application A_i and $R_i \cap R_j = \varnothing$ for $\forall i \neq j$. Note that applications can be run concurrently in any combination; a Resource allocation mechanism within separate applications is not specified.

3) RERC-2: Radio Applications use pre-defined static Resources.

RERC-2 implements multiple Radio Applications but no dynamic Resource management is available. The Radio Applications for RERC-2 come from a single Radio Application Package which is normally provided by a reconfigurable RE vendor or SDR chipset manufacturer. In this scenario, it is assumed that software radio components in the Radio Application Package are provided in platform-specific executable code.

The rule for the Resource allocation related to multiple applications $\{A_1, A_2, \ldots, A_N\}$ can be formulated as follows: $A_i \rightarrow R_i$, $\forall i \in \{1, ..., N\}$, where R_i denotes Resources allocated for application A_i, if $\exists\, i \neq j$ so that $R_i \cap R_j \neq \varnothing$ then such applications cannot be run concurrently, all other combinations are allowed; a Resource allocation mechanism within separate applications is not specified.

4) RERC-3: Radio Applications have static Resource requirements.

For RERC-3, a Resource budget is defined for each Radio Application. This budget contains a static Resource measure that represents the worst-case Resource usage of the application, generated at Radio Application compile-time. If an application is being started, the Resource manager installed in a reconfigurable RE of RERC-3 checks its Resource budget and the sum of all Resource budgets of already running applications, and admits the new application only if the Resources can still be guaranteed for all running applications. In this scenario, it is assumed that software radio components in the Radio Application Package are provided in platform-specific executable code.

The rule for Resource allocation for multiple applications $\{A_1, A_2, \ldots, A_N\}$ can be formulated as follows:

$A_i \rightarrow \mathrm{R}(A_i)$, where R denotes total Resources to be shared and $\mathrm{R}(A_i)$ denotes a part of R allocated for A_i; if for $i1, i2, ..., iM \in \{1, ..., N\}$, $M \leq N$, $\mathrm{R}(A_{i1}) \cup \mathrm{R}(A_{i2}) \cup ... \cup \mathrm{R}(A_{iM}) \subset \mathrm{R}$ then applications $A_{i1}, A_{i2}, \ldots, A_{iM}$ can be run concurrently; a Resource allocation mechanism within separate applications is not specified.

5) RERC-4: Radio Applications have dynamic Resource requirements.

This scenario assumes a similar Resource manager in a reconfigurable RE as for MDRC-3, but in addition the Radio Applications have now varying Resource demands based on their current type of activity. Applications have separate operational states for different types of activity, and a Resource budget is assigned to each operational state. In this scenario, it is assumed that software radio components in the Radio Application Package are provided in platform-specific executable code.

Resource management for RERC-4 can be formulated as follows. Multiple applications $\{A_1, A_2, \ldots, A_N\}$ can be run and each application A_i is divided into tasks $\{t_1(A_i), t_2(A_i), ..., t_k(A_i)\}$. Resource allocation is provided by the Resource manager in a reconfigurable RE for each task $t_j(A_i) \rightarrow \mathrm{R}(t_j(A_i))$.

The rule for task running is exactly the same as for RERC-3 except that each application should be replaced by a corresponding task. Therefore, if for $i1, i2, ..., iM \in \{1, ..., N\}$, $M \leq N$, $\mathrm{R}(t_{j1}(A_{i1})) \cup \mathrm{R}(t_{j2}(A_{i2})) \cup ... \cup \mathrm{R}(t_{jL}(A_{iM})) \subset \mathrm{R}$ then tasks $t_{j1}(A_{i1}), t_{j2}(A_{i2}), ..., t_{jL}(A_{iM})$ can be run concurrently; a Resource allocation mechanism within separate tasks is not specified.

6) RERC-5: Radio Applications use pre-defined static Resources, on-device compilation of Software Radio Components.

This class corresponds to RERC-2 with the difference that all or part of the software radio components are provided in the Radio Application Package as platform-independent source code or platform-independent Intermediate Representation (IR), which is compiled on the reconfigurable RE itself. It particularly means that the reconfigurable RE should include a proper compiler in order to convert the source code or IR of the software radio components into an executable code that runs on a given modem chip of a reconfigurable RE. It is assumed that the methods of radio programming and the tools to support this category have become sufficiently standardized so that third-party vendors may create Radio Applications and activate them to different platforms with relative ease. The formal description of the Resource management is the same as for RERC-2.

7) RERC-6: Radio Applications have static Resource requirements, on-device compilation of Software Radio Components.

This class corresponds to RERC-3 with the difference that all or part of the software radio components are provided in the Radio Application Package as platform-independent source code or platform-independent IR, which is compiled on the reconfigurable RE itself. As in the case of RERC-5, it particularly means that the reconfigurable RE should include a proper compiler in order to convert the source code or IR of the software radio components into an executable code that runs on a given modem chip of a reconfigurable RE. As in the case of RERC-5, it is assumed that the methods of radio programming and the tools to support this category have become sufficiently standardized so that third-party vendors may create Radio Applications and activate them to different platforms with relative ease. The formal description of the Resource management is the same as for RERC-3.

8) RERC-7: Radio Applications have dynamic Resource requirements, on-device compilation of Software Radio Components.

This class corresponds to RERC-4 with the difference that all or part of the software radio components are provided in the Radio Application Package as platform-independent source code or platform-independent IR, which is compiled on the reconfigurable RE itself. As in the case of RERC-5 or RERC-6, it particularly means that the reconfigurable RE should include a proper compiler in order to convert the source code or IR of the software radio components into an executable code that runs on a given modem chip of a reconfigurable RE. As in the case of RERC-5 or RERC-6, it is assumed that the methods of radio programming and the tools to support this category have become sufficiently standardized so that third-party vendors may create Radio Applications and activate them to different platforms with relative ease. The formal description of the Resource management is the same as for RERC-4.

The definition of RERCs described above can be summarized as shown in Table 1.

Table 1: Summary of RERCs

<table>
<tr><th></th><th>Multi-radio system</th><th>Resource Share (among Radio Applications)</th><th>Resource Manager</th><th>Multi-tasking</th><th>Resource Measurement</th><th>Resource Allocation</th></tr>
<tr><td>RERC-0</td><td>No</td><td>No</td><td>No</td><td>No</td><td>Design-time</td><td>Design-time</td></tr>
<tr><td>RERC-1</td><td>Yes</td><td>No</td><td>No</td><td>No</td><td>Design-time</td><td>Design-time</td></tr>
<tr><td rowspan="2">RERC-2
RERC-5</td><td rowspan="2">Yes</td><td rowspan="2">No
(note 1)</td><td rowspan="2">Yes
(note 2)</td><td rowspan="2">Yes
(note 3)</td><td>Design-time</td><td>Design-time</td></tr>
<tr><td>Design-time
/Install-time</td><td>Design-time
/Install-time</td></tr>
<tr><td rowspan="2">RERC-3
RERC-6</td><td rowspan="2">Yes</td><td rowspan="2">Yes</td><td rowspan="2">Yes</td><td rowspan="2">Yes</td><td>Design-time</td><td rowspan="2">Run-time</td></tr>
<tr><td>Design-time
/Install-time</td></tr>
<tr><td rowspan="2">RERC-4
RERC-7</td><td rowspan="2">Yes</td><td rowspan="2">Yes</td><td rowspan="2">Yes</td><td rowspan="2">Yes</td><td>Design-time</td><td rowspan="2">Run-time</td></tr>
<tr><td>Design-time
/Install-time</td></tr>
<tr><td colspan="7">NOTE 1: Resource share can exist among Radio Access Technologies (RATs) in a given Radio Application.
NOTE 2: This is for a fixed Resource allocation only. Resource management and Resource allocation among RATs (in a single RA) are pre-determined in a static manner by Radio Application provider.
NOTE 3: Multi-tasking in this case is for multiple RATs within a single Radio Application.</td></tr>
</table>

Note that radio conformance tests are mandatory for RERC-1 to RERC-7 in order to ensure that the joint operation of (dynamically) reconfigured base-bands and RF front-ends are in compliance with the relevant conformance requirements before the device is introduced into the market.

The requirements described in the present document are based on the RERCs above defined. As it can be noted in clause 6, some requirements are independent from the class of device while others apply only to well defined classes. Therefore a reconfigurable RE will follow only those requirements related to the RERC it belongs to.

6 Functional Requirements

6.1 Requirements on RAT Link Support and Management

6.1.1 R-FUNC-RAT–01 Function for RERC-1 to RERC-7

A reconfigurable radio equipment should support parallel connections to more than one Radio Access Technology.

6.1.2 R-FUNC-RAT–02 Function for RERC-1 to RERC-7

If a reconfigurable radio equipment allows parallel connections to RATs (in alignment to R-FUNC-RAT-01), in-device coexistence functionalities shall be implemented.

6.1.3 R-FUNC-RAT–03 Function for RERC-1 to RERC-7

If a reconfigurable radio equipment allows parallel connections to RATs (in alignment to R-FUNC-RAT-01), seamless handover of data streams from one RAT to another RAT should be implemented.

NOTE: Seamless handover is only between RATs used for the same service.

Explanation: A seamless handover does not create any interruption of an ongoing service.

6.1.4 R-FUNC-RAT–04 Function for RERC-1 to RERC-7

If policies are applied to a reconfigurable radio equipment, the link selection functionality in the reconfigurable radio equipment shall meet the related conditions.

Explanation: It is possible to provide policies to reconfigurable radio equipment. These policies introduce link selection constraints to be met by the link selection decision making functionality in the concerned reconfigurable radio equipment. For example, the policies may be used for enforcing Network Operator preferences, User preferences, etc.

6.1.5 R-FUNC-RAT–05 Function for RERC-1 to RERC-7

If a reconfigurable radio equipment allows parallel connections to RATs (in alignment to R-FUNC-RAT–01), various independent data flows should be maintained simultaneously.

6.1.6 R-FUNC-RAT–06 Function for RERC-1 to RERC-7

If a reconfigurable radio equipment allows parallel connections to RATs (in alignment to R-FUNC-RAT–01), Link Adaptation techniques across multiple RATs should be implemented.

NOTE: Link Adaptation techniques include Network Coding, Air Interface selection, etc.

6.2 Radio Application Requirements

6.2.0 General

The following requirements are based on the Use Cases described in ETSI TR 102 944 [i.2].

6.2.1 R-FUNC-RA-01 Radio Applications Support for RERC-1 to RERC-7

Reconfigurable radio equipment shall support the execution of Radio Applications.

6.2.2 R-FUNC-RA-02 Composition for RERC-1 to RERC-7

Radio Applications shall be composed of SFBs and/or UDFBs.

Explanation: The resulting Radio Applications will be composed by combining SFBs and/or UDFBs. They will be used, for example, for baseband signal processing and real-time Media Access Control (MAC) data processing.

6.2.3 R-FUNC-RA-03 Concurrency for RERC-1 to RERC-7

Reconfigurable radio equipment should support concurrent execution of Radio Applications.

6.2.4 R-FUNC-RA-04 Data for RERC-1 to RERC-7

Radio Applications should support the function of transferring receive (Rx)/transmit (Tx) data to/from the networking stack.

6.2.5 R-FUNC-RA-05 Context Information for RERC-1 to RERC-7

Radio Applications should support the function of delivering context information.

Explanation: Access to context information metrics is provided. As an example, the metrics can be related to:

- Signal to Interference-plus-Noise Ratio (SINR)
- Received Signal Strength Indication (RSSI)
- Packet Error Rate (PER)
- Bit Error Rate (BER)
- Power consumption of selected Base-Band modules
- MIMO algorithm configuration indicator
- Precoding Matrix Indicator (PMI)
- Rank Indicator (RI)
- Data rate indicator
- Channel Coefficients Information

6.2.6 R-FUNC-RA-06 Pipelining for RERC-2 to RERC-7

Radio Applications shall be applicable to fixed pipeline, programmable pipeline, and hybrid pipeline architectures.

Explanation: The pipeline of functional blocks is determined by the contents of metadata included in the Radio Application Package. Modem chip manufacturer selects one of the 3 pipeline structures for their modem chip processor. When the modem chip processor is configured according to the contents of metadata, the configuration of modem chip processor is performed according to the pipeline structure adopted in a given modem chip.

6.3 Radio Application Functional Block Requirements

6.3.1 R-FUNC-FB-01 Implementation for RERC-2 to RERC-7

Each instance of a functional block shall be implemented with a corresponding program code characterized by the implementation properties and shall be accessed by a corresponding interface.

Explanation: The typical structure of the functional block is shown in Figure 3. The corresponding interface is represented as "*Name(Input, Output)*" and the functionality of each functional block is implemented with the corresponding program code attributed by the properties.

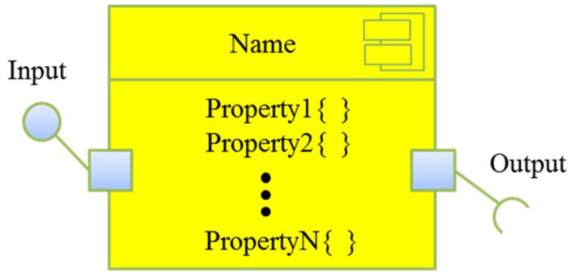

Figure 3: Functional block defined as a baseband signal processing component

6.3.2 R-FUNC-FB-02 Execution for RERC-2 to RERC-7

Each instance of a functional block shall be executed only by calling the corresponding interface.

Explanation: The execution of an instance of a functional block requires an external trigger, for example in order to avoid recursion.

6.3.3 R-FUNC-FB-03 Side Effects for RERC-2 to RERC-7

The internal state of an instance of each functional block shall not be shared with any other instances of functional blocks.

Explanation: The behaviour of an instance of a functional block depends only on its inputs and not on the state of another instance of a functional block.

6.3.4 R-FUNC-FB-04 Shared Data for RERC-2 to RERC-7

The reconfigurable radio equipment should support data sharing with other instances of functional blocks.

Explanation: Input and/or output of each functional block can be shared with other instances of functional block(s).

6.3.5 R-FUNC-FB-05 Concurrency for RERC-2 to RERC-7

The reconfigurable radio equipment should support the concurrent execution of instances of functional blocks.

6.3.6 R-FUNC-FB-06 Extendability for RERC-2 to RERC-7

A library extension shall be supported.

Explanation: A Radio Application consists of UDFBs as well as SFBs. Meanwhile, the Radio Library of a reconfigurable radio equipment consists of SFBs only and not of UDFBs. Therefore, the reconfigurable radio equipment needs a normative library extension to support UDFBs. In other words, the reconfigurable radio equipment needs to support a normative procedure for a normative library extension.

In case of RERC-5 to RERC-7, the library extension is performed on the reconfigurable radio equipment.

6.4 Radio Equipment Reconfiguration Requirements

6.4.1 R-FUNC-RER-01 Platform-specific Executable Code for RERC-2, RERC-3 or RERC-4

The configuration of a reconfigurable radio equipment compliant to RERC-2, RERC-3 or RERC-4 shall be realized with a Radio Application Package of which the user defined functional blocks, if any, are provided in platform-specific executable code.

Explanation: A Radio Application Package provided with the platform-specific executable code is designed specifically for each kind of target hardware platform. Each platform-specific executable code is generated using a specific compiler for the corresponding target hardware platform during design time. Then, the reconfigurable radio equipment configuration is performed through downloading of the compiled code into the reconfigurable radio equipment and its installation.

6.4.2 R-FUNC-RER-02 Platform-independent Source Code or IR for RERC-5, RERC-6 or RERC-7

The configuration of a reconfigurable radio equipment compliant to RERC-5, RERC-6 or RERC-7 shall be realized with a Radio Application Package of which the user defined functional blocks, if any, are provided either in a platform-independent source code or an Intermediate Representation (IR).

Explanation for Source Code case:

A Radio Application Package can adopt user defined functional blocks in a form of platform-independent source code with a proper encryption. Since a Radio Application Package is provided in platform-independent source code, the user defined functional block code is compiled to generate corresponding executable code using a compiler for the specific reconfigurable radio equipment hardware platform. Note that standard functional blocks are based on a reference design given by the Radio Library that is given in platform specific executable code.

Explanation for IR case:

A Radio Application Package adopting a platform-independent IR is targeted to different kinds of modem chips. Since a Radio Application Package is provided in a platform-independent IR (i.e. non-executable code), the platform-independent IR is always translated into executable code using a back-end compiler for the specific reconfigurable radio equipment hardware platform. Note that standard functional blocks are based on a reference design given by the Radio Library that is given in platform specific executable code.

6.4.3 R-FUNC-RER-03 Radio Configuration of Platform RERC-1 to RERC-7

The radio configuration of a reconfigurable radio equipment shall be realized with the activation of Radio Applications (RA) and, if necessary, changing parameters of the activated RAs.

6.4.4 R-FUNC-RER-04 Radio Programming for RERC-1 to RERC-7

The reconfigurable radio equipment shall provide a suitable interface which conveys structural and behavioural information of RAs for the reconfigurable radio equipment reconfiguration.

Explanation: The interface enables the provision of reconfiguration code onto the reconfigurable radio equipment. The structural information defines a set of functional blocks or RA computational operators, data and communications among them. The behavioural information defines execution rules and operator interactions.

6.4.5 R-FUNC-RER-05 Dynamic Execution for RERC-4, and RERC-7

The reconfigurable radio equipment shall support dynamic execution of functional blocks.

Explanation: In case of dynamic Resource sharing, the Resource allocation is performed in run time.

6.4.6 R-FUNC-RER-06 Independency on Memory Model for RERC-1 to RERC-7

Different radio platform memory models shall be supported. Those shall include the shared memory and the message passing memory models.

6.4.7 R-FUNC-RER-07 Code for RERC-2 to RERC-7

The interface related to R-FUNC-RER-04 shall support the provision of executable code (for RERC-2 to RERC-4) and IR and/or Source Code (for RERC-5 to RERC-7).

6.4.8 R-FUNC-RER-08 IR Format for RERC-5 to RERC-7

IR shall be defined in a format suitable for human reading/writing and automated processing.

6.4.9 R-FUNC-RER-09 Timing Constraints for RERC-1 to RERC-7

The interface related to R-FUNC-RER-04 shall support timing constraints for radio processing.

Explanation: This requirement relates to real time execution. The execution of concerned functional blocks will be in accordance to the provided timing constraints.

6.4.10 R-FUNC-RER-10 Platform Independency for RERC-5 to RERC-7

The reconfigurable radio equipment architecture shall provide suitable interfaces in order to ensure platform independency.

6.4.11 R-FUNC-RER-11 Radio Application for RERC-5 to RERC-7

The Software representation (i.e. IR or Source Code) of an RA shall be independent of the target hardware platform.

Explanation: The independence of the RA from the target hardware platform indicates that the RA can be provided to multiple platforms supporting the interfaces related to R-FUNC-RER-10.

6.4.12 R-FUNC-RER-12 Function Granularity for RERC-1 to RERC-7

Functional blocks may have different granularity.

Explanation: The functional blocks might be for example arithmetic operations or functions like FFT or even whole Radio Applications like WiFi, LTE, etc.

6.4.13 R-FUNC-RER-13 Radio Virtual Machine for RERC-2 to RERC-7

Radio Application(s), SFB(s) or UDFB(s) shall be executed on a suitably configured Radio Virtual Machine, including the application of a suitable protection class.

Explanation: A RVM execution environment should be provided in such a way that a proper RVM (protection) class can be selected by each vendor. The RVM approach is required for platform dependent and/or independent 3rd party code, since a manufacturer will require that 3rd party code is executed in a controlled environment. The 3rd party code corresponds to a Radio Application, SFB(s) or UDFB(s).

6.4.14 R-FUNC-RER-14 Radio Virtual Machine Structure for RERC-2 to RERC-7

A Radio Virtual Machine may consist of several smaller Radio Virtual Machines.

Explanation: There may be an RVM hierarchy, i.e. one or several RVMs might be a part of the bigger RVM. A smaller RVM is connected to external RVM ports of the bigger RVM.

6.4.15 R-FUNC-RER-15 Selection of Radio Virtual Machine Protection Class for RERC-2 to RERC-7

A Radio Application shall select a suitable Radio Virtual Machine Protection Class.

Explanation: Radio Virtual Machine (protection) classes are introduced in order to find a trade-off between (re-) certification effort and base-band code development flexibility.

At one extreme of RVM class, a high-level RVM class corresponds to full reconfigurability of the low-level parameters of an RVM, and accordingly necessitates a relatively more extensive certification testing process after the RVM has been reconfigured. At the other extreme of RVM class, a low-level RVM class corresponds to a limited reconfigurability of the low-level parameters of an RVM. As the reconfigurability of the low-level parameters of this particular class of RVM is limited, a relatively less extensive certification testing process is necessitated after the RVM has been reconfigured. Moreover, an RVM can have different RVM classes associated with different components of the RVM that relates to the reconfigurability of the low-level parameters of the respective components of the RVM.

Reconfiguration of an RVM of the highest-level RVM class may necessitate that the overall certification testing process focuses on the certification of each reconfigured software components of the RVM. In such a situation, each respective reconfigured software component may need to be separately certified before one or more sets of reconfigured software components are certified together. For example, a reconfigured RVM software component "A" (e.g. WiFi) may need to be separately certified from a reconfigured and certified RVM software component "B" (e.g. LTE). The certification process may then be such that the joint operation of separately certified reconfigured RVM software components "A" and "B" may then take place jointly.

At the other extreme of RVM classes, the lowest-level RVM class corresponds to a restricted reconfigurability of the low-level parameters of an RVM. For such a restricted level of reconfigurability, a developer of Radio Applications would only have limited access to the low-level parameters of an RVM. For example, the lowest-level RVM class would permit a Radio Application developer to have access to only the low-level parameters of the receive chain of an RVM. Accordingly, the lowest-level of RVM class would not need to utilize a corresponding detailed and thorough certification testing process because, for example, a radio platform operating a malfunctioning reconfigured RVM would not interfere with other radio platforms. Thus, level of certification testing for the lowest RVM class would be less extensive certification testing process than that used for the highest RVM class.

One or more medium-or intermediate-level RVM classes may also be established between the two extreme RVM classes that correspond to intermediate levels of reconfigurability of the low-level parameters of an RVM. An intermediate-level RVM class, for example, would allow more flexibility for reconfiguring low-level parameters of an RVM than the lowest-level RVM class, but would not permit the degree of reconfigurability that would be associated with the highest-level RVM class. Depending on the level of reconfigurability to the low-level parameters of an RVM, an intermediate-level RVM class may necessitate a certification testing process for a compiled reconfigured RVM and underlying hardware that is more extensive than that corresponding to the lowest-level RVM class, but less extensive than that corresponding to the highest-level RVM class. For example, a certification based on the intermediate-level RVM software component might be obtained by contacting an authorized notified body and providing only a serial number for the RVM software component and an identification of the target device type on which the compiled reconfigured RVM would operate. In another example, there could be no requirement for a joint certification based on an RVM software component for a simultaneous operation with other RVM software components. That is, a certificate based on an RVM software component "A" (e.g. WiFi) and a separate certificate based on another RVM software component "B" (e.g. LTE) would allow for a simultaneous operation of reconfigured components "A" (e.g. WiFi) and "B" (e.g. LTE).

Another exemplary situation that may necessitate a relatively less extensive certification testing process would be a Radio Application developer that only reconfigures non-transmission-related low-level parameters, for example, low-level parameters relating to a data interleaver and/or a channel coder in the transmit/receive (TX/RX) chain of an RVM that otherwise has been defined to be of the highest-level RVM class. As nothing related to the spectral shaping of a transmitted signal is reconfigured by the reconfiguration of the data interleaver and/or channel coder, a relatively less extensive certification testing process could be used. Another exemplary situation that may necessitate a less extensive certification testing process would be a reconfiguration that involves changes targeting predefined frequency bands and/or bandwidths. In still other exemplary situations, there may be reconfigurations for which a certification testing process may not be necessary.

6.4.16 R-FUNC-RER-16 Distributed Computations for RERC-5, RERC-6, RERC-7

The reconfigurable radio equipment shall support distributed computations in the case when the radio platform of the reconfigurable radio equipment consists of a number of processors each of which is distributed across different physical entities.

NOTE: Different physical entities may be located in close proximity or remotely.

6.5 Radio Frequency (RF) Transceiver Requirements

6.5.0 General

The following requirements are related to the RF transceiver which allows for the selection of RF configuration parameters. However, it does not limit the manufacturer in the choice of supported features in a given platform.

6.5.1 R-FUNC-RFT-01 RF Configuration for RERC-1 to RERC-7

The reconfigurable radio equipment shall provide a suitable interface for RF transceiver configuration.

Explanation: The interface will enable the exchange of control and data information between the Radio Applications and the RF transceiver.

6.5.2 R-FUNC-RFT-02 Extendibility for multiple-antenna system for RERC-1 to RERC-7

If a reconfigurable radio equipment supports multiple antenna operation, a Radio Application shall be able to select a suitable number of antenna inputs/outputs.

Explanation: The 3rd party developers can choose among multiple antenna technologies such as, for example, SU-MIMO, MU-MIMO, massive MIMO, etc. The extendibility of physical antennas relates to the number of RF input/output signals from/to the RF front-end(s) to to/from the antennas.

6.5.3 R-FUNC-RFT-03 Capability of multiple frequency bands for RERC-1 to RERC-7

The reconfigurable radio equipment shall support multiple Radio Applications using distinct frequency bands.

6.5.4 R-FUNC-RFT-04 Reconfigurability of RF Transceiver for RERC-1 to RERC-7

An RF transceiver shall manage input/output signals from/to one or several Radio Applications.

Explanation: Several Radio Applications which are simultaneously in active state may be served by one or multiple RF transceivers.

6.5.5 R-FUNC-RFT-05 Interoperability of radio resources for RERC-2 to RERC-7

Sharing of radio resources among multiple Radio Applications shall be supported.

6.5.6 R-FUNC-RFT-06 Testability of radio equipment for RERC-1 to RERC-7

A test mode in which the transmitter chain is connected to the receiver chain in loop-back mode shall be supported.

Explanation: The test mode provides test capability of the RF path without actual radio waves emission.

6.5.7 R-FUNC-RFT-07 Unified representation of control information for RERC-1 to RERC-7

The interface related to R-FUNC-RFT-01 shall support a unified representation of control information passed to/from the RF front-end.

Explanation: Control information passed to/from the RF front-end is represented in a unified format suitable for RF front-end handling.

6.5.8 R-FUNC-RFT-08 Unified representation of data payload for RERC-1 to RERC-7

The interface related to R-FUNC-RFT-01 shall support a unified representation of data payload passed to/from the RF front-end.

Explanation: Data payload passed to/from the RF front-end is represented in a unified format suitable for RF front-end data handling.

6.5.9 R-FUNC-RFT-09 Selection of RF Protection Class for RERC-1 to RERC-7

The interface related to R-FUNC-RFT-01 shall support a suitable selection of an RF Protection Class.

Explanation: RF protection classes are introduced in order to find a trade-off between (re-)certification effort and RF front-end flexibility. In case of a low RF Protection Class, the RF front-end provides a high level of flexibility (e.g. in terms of band selection, bandwidth selection, out-of-band radiation, etc.) which typically requires a thorough and complex (re-)certification of the concerned Radio Applications. In case of a high RF Protection Class, on the other hand, the RF front-end typically introduces protection mechanisms (e.g. filters for limiting out-of-band radiation, etc.) which limit the flexibility of the RF front-end but allow for a lighter (re-)certification of corresponding Radio Applications.

6.6 Security Requirements

6.6.0 General

The following requirements are related to End-to-End Security which allows to ensure that only approved Radio Applications may be downloaded to and executed on a target Radio Equipment.

6.6.1 R-FUNC-SEC-01 REConfPol-RAP-Security

Cryptographic functions shall be provided to support the digital signature strategy providing confidentiality, integrity, and authenticity of RE Configuration Policy and RAP(s), and the non-repudiation strategy.

6.6.2 R-FUNC-SEC-02 Administration-Security

A security function in the Communication Service Layer shall ensure that only approved RAP(s) are processed and executed on a target platform.

Explanation: Typically, an *Administrator Security Function* is added to the *Administrator* entity in the *Communication Service Layer*.

6.6.3 R-FUNC-SEC-03 Secure Management

Security functions shall be provided to allow remote parties to securely attest to the status of the Radio Equipment, configure the Radio Equipment, and manage it over extended period of time.

Explanation: Typically, an *Attesting Entity* is present on the device (possibly in a distributed fashion), that can act as a root of trust for reporting. The *Administrator* entity in the *Communication Service Layer* is able to receive, validate, and act on configuration commands from authorized remote parties.

The flexibility introduced by radio reconfiguration will greatly extend the lifetime of the device. It shall be possible to bind the device to a new configuration authority when necessary. Typically, an *RRS Configuration Manager* is added to the *Administrator* entity in the *Communication Service Layer* for the purpose of keeping track of active configuration authorities.

6.6.4 R-FUNC-SEC-04 Root of Trust

A root of trust shall provide security services to the reconfigurable platform.

Explanation: components are required in order to provide a high level of security assurance throughout the lifecycle of the Radio Equipment.

History

Document history		
V1.1.1	March 2019	Publication