

# IDROMe1

## Impact des équipements reconfigurables pour le déploiement des futurs réseaux mobiles

Projet de plateforme RNRT 2005

### Livrable #1.2

#### Cahier des charges sommaire du démonstrateur



En collaboration avec

Référence	2006_06_30-IDROMel-deliv1.2-0.1
Auteur responsable	CEA/Leti
Partenaires	Eurécom, Thales, Supélec, Telecom Paris, CEA, France Télécom R&D, Siradel, ANFR
Date	30/09/2006
Type	LIVRABLE
Identifiant	1.2
Version	1.0
Statut	version finale
Groupe	RNRT
Diffusion	IDROMel
Résumé	
Mots clefs	livrable 2, cahier des charges, IDROMel

## HISTORIQUE DES CHANGEMENTS

Version	Date	Author	Chapitres	Modifications
0.1	21/07/2006	Jérôme MARTIN Tous partenaires	Tous	Création du document Intégration des contributions des partenaires
0.2	22/08/2006	Dominique NUSSBAUM		Intégration contributions RF/Phy/L2
0.3	13/09/2006	Renaud PACALET Jérôme MARTIN	2 et 3	Intégration contribution ENST Mise en forme, ajout 2.3.2 et conclusion
0.4	18/09/2006	Bertrand MERCIER	2.4 et 2.5.1	Contribution sur Protocol stacks et Configuration Management
0.5	25/09/2006	Jérôme MARTIN		Suppression §2.6 et mise en forme

1	Introduction .....	4
2	Requirements and functionalities implied by the scenarios on the IDROMel demonstrator .....	5
2.1	Mobile Terminal Functionalities and Architecture .....	5
2.2	RF Front End .....	7
2.3	Physical Layer .....	9
2.3.1	ADAC/Baseband Subsystem .....	10
2.3.2	FAUST board.....	12
2.4	Protocol Stacks .....	12
2.4.1	UMTS MAC-Layer Architecture .....	13
2.4.2	Ad-hoc MAC layer Architecture .....	15
2.5	Reconfiguration Management .....	16
2.5.1	Protocol stack reconfiguration architecture.....	16
2.5.2	PHY/RF reconfiguration management .....	17
3	Conclusions.....	21
4	Reference .....	22

# 1 Introduction

The IDROMEL RNRT project aims at:

- Developing reconfigurable (Software Defined Radio – SDR) equipment able to communicate through at least 2 Radio Access Technologies (RAT) using two different frequency bands;
- Providing a wide access to our results for the R&D community (French and International).

In IDROMel deliverable #1.1 ( [1]) several use cases of reconfigurable radio systems have been presented, based on state-of-the-art technologies in this domain. Some SDR key concepts have been inferred from these use cases, which should be demonstrated in the frame of the IDROMel project:

- Seamless Vertical Handover: this concept illustrates the possibility for a wireless terminal to switch from one RAT to another without interrupting any network service currently used by the terminal. Therefore this switch should be invisible to the terminal user (except possibly an adaptation of the QoS to the new RAT's bandwidth);
- Reconfiguration Management: the complexity induced by terminal reconfiguration abilities — in particular the vertical handover, but also other reconfiguration scenarios like transparent bug correction or protocol stack improvements — requires a specific effort to demonstrate the feasibility of dealing with such a complex system.
- Cognitive Radio: in the most liberal sense this concept describes the ability of the radio system to react to its environment. A classical example is a radio system able to detect unused frequency bands in the radio spectrum, which can therefore select the best RAT and/or bands to communicate in its environment. However a lot of other external parameters, possibly including network services' requests, may be used to trigger a new behaviour of the mobile terminal.

One can notice that these three key-concepts should be very tightly related in a high-performance SDR system: a change in the mobile terminal's environment may be detected and analysed by the system, which decides to trigger a vertical handover to a better suited RAT, and the reconfiguration management system realizes necessary hardware and software reconfiguration to effectively perform this handover.

Taking into account all these elements, 3 demonstration scenarios (A, B and C, described below) have been derived in [1], which could be demonstrated in the frame of IDROMel. They will be demonstrated in a graduate way: it means that a first demonstration will be based on scenario A, then we will demonstrate the scenario B and finally depending on the progress within the project the scenario C may be demonstrated.

- A. UMTS and WiMax running on the same platform, but separately, i.e. the one after the other without service continuity (2 separated MAC). A SISO implementation is planned at this stage. This scenario will prove the feasibility of two RATs running simultaneously on a terminal, which corresponds to the intermediate phase of a vertical handover, when a new MAC has been set up while previous RAT is still running, ensuring continuity of service.
- B. WiMax in a MIMO 2x2 configuration. This scenario describes the terminal situation before (or after) a vertical handover, thus constituting a starting point (or. ending point) to realise a vertical handover. Scenario B may be implemented before scenario A.

- C. This scenario plans a soft handover from UMTS to WiMax or WiMax to UMTS with a switch at the MAC level. The realisation of this scenario will depend on the progress within the project.

The purpose of this document is to infer from these three demonstration scenarios the functionalities and requirements that have to be fulfilled by the equipment. We will first present the global mobile terminal specificities needed to realize the demonstration scenarios, and their consequences on MT architecture. Then we will go further into details in the needed functionalities in term of RF front-end, physical layer implementation, protocol stacks, reconfiguration management, and finally have a look at the cognitive radio aspects in the IDROMel demonstrator.

## 2 Requirements and functionalities implied by the scenarios on the IDROMel demonstrator

### 2.1 Mobile Terminal Functionalities and Architecture

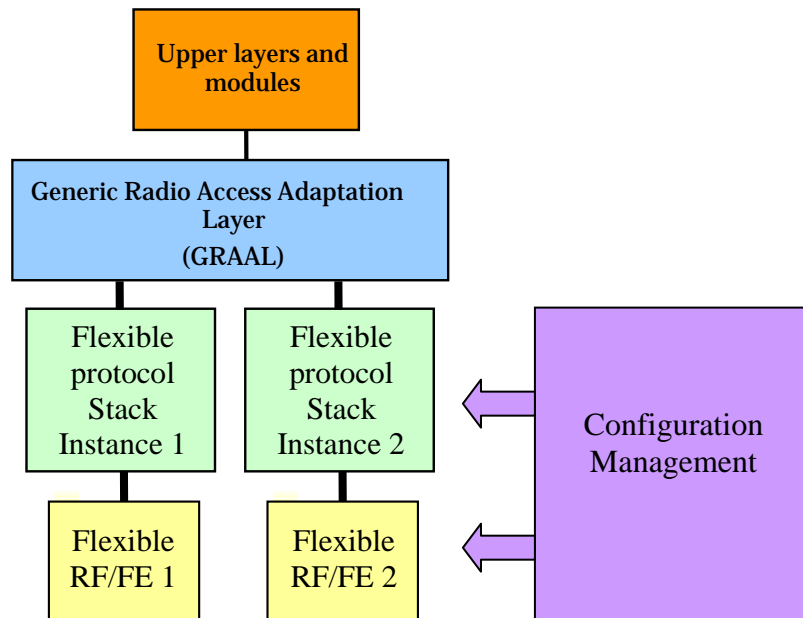
From the mobile point of view, its architecture must be designed for the validation of SDR scenarios. The aim of our work is to enable a terminal to move seamlessly in a heterogeneous network, including at least two different RATs (with different QoS parameters, frequency bands and bandwidths). The selected RATs in the frame of IDROMel are UMTS and WiMax, but as far as possible the developed terminal will be able to deal with other RAT, thanks to its reconfiguration abilities.

Moreover, we target a quasi-optimal handover: optimality means that no degradation due to handover is observed. This quasi-optimal handover will be possible thanks to a Duplication and Merging approach developed in E<sup>2</sup>R ( [2]) and Rhodos. The basic idea of this approach is to send the data flows from and to the mobile through 2 radio interfaces at the same time during the handover. In order to achieve this *soft handover*, there is a clear need of 2 RF front-ends and two entities of the protocol stack working in parallel.

The global architecture (hardware and software) of the mobile equipment is designed in order to fulfil the following functionalities:

- working in real-time,
- including the necessary protocol stack that enable the communication with a basestation or an access point,
- ability to communicate through 2 different RATs with 2 different bandwidths and frequency bands,
- reconfigurability of each RAT, including physical Layer,
- ability to switch seamlessly from a RAT to another.

The resulted functional overall architecture is depicted in Figure 1.



**Figure 1: Functional Architecture for a Double RAT Transmission**

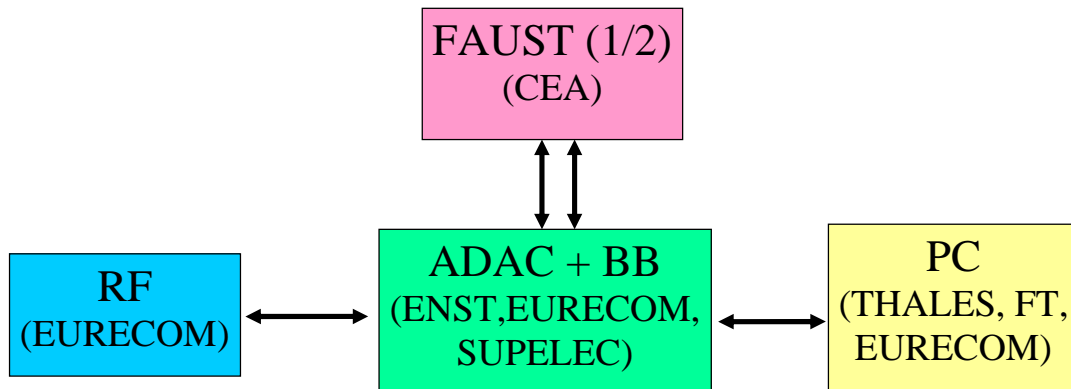
In this architecture, the GRAAL (Generic Radio Access Adaptation Layer) module aims at hiding the heterogeneity of the RATs (QoS parameters for example) to the upper layers. Moreover, the 2 instances of protocol stack and Radio Frequency Front-End (RF/FE) allow the mobile terminal either to communicate through 2 different RATs during a vertical handover, either to communicate through one standard using multiple antennas processing (MIMO).

To enable those global functionalities — 2 instances for the protocol stack and RF — we need some specific local functionalities such as:

- flexible or multi-band antennas,
- flexible RF Front-End,
- reconfigurable base-band,
- reconfigurable protocol stack,
- logical entity able to handle communication through 2 different protocol stacks and interface the upper layers.

Clearly, we will focus on the lower layers (from the antennas to the L2) in the frame of IDROMel.

From a practical point of view, the demonstrator architecture should be as flexible/modular as possible, in order to limit the risks. Hence, the foreseen overall architecture and interfaces for the Mobile Terminal are depicted in the figure below.



**Figure 2: Foreseen HW Architecture**

As depicted in the above figure, the architecture comprises 4 modules:

- a RF board that will include 4 TX, 4 RX RF chains (MIMO 4\*4), and will cover 400 MHz to 7 GHz. It has an analog (for Rx and Tx signals, I/Q at low IF or BB) and a digital (control of the LNAs, switches, PA....) interface with the rest of the system;
- a ADAC + BB board : this board has a central role in the system since it interfaces all the others components. This board includes 4 ADAC and 2 powerful FPGAs;
- a baseband daughter board built around the FAUST chip, bringing its baseband computing and reconfiguration capabilities to the system;
- a PC implementing upper protocol stacks.

In order to minimize the risks, it will be possible to use the modem without the CEA part (but with reduced functionalities). Regarding the FAUST chip (FAUST 1 in the first version of the daughter board, FAUST 2 in a second version), it interfaces the ADAC+BB board. This interface is purely digital and depends strongly on the targeted functionalities.

## 2.2 RF Front End

Concerning the Radio Frequency, it should allow the validation of the key concepts and scenarios developed in the IDROMel project. From a RF point of view, the main objective of the proposed prototype is to show that a highly reconfigurable RF transceiver is possible with existing available components.

### A. Key features

First of all, the targeted prototype (see Figure 3) is very ambitious in term of frequency bands, since the objective is to address from 400 MHz to 7.5 GHz, with a maximum bandwidth of 20 MHz. Hence, we will be able to receive and transmit almost all the existing commercial Radio Access Technologies. Concerning the transmitted power, the target is comparable to existing GSM terminals (+21 dBm). On the receiver side, the objective is to have a noise figure from 8 to 12 dB, depending on the frequency band. Since IDROMel is also considering Multiple Antenna Processing, the RF equipment will include up to 4 antennas and 4 RF chains. Finally, three key features of the targeted prototype are:

- It will integrate advanced re-sampling functionalities as described previously.

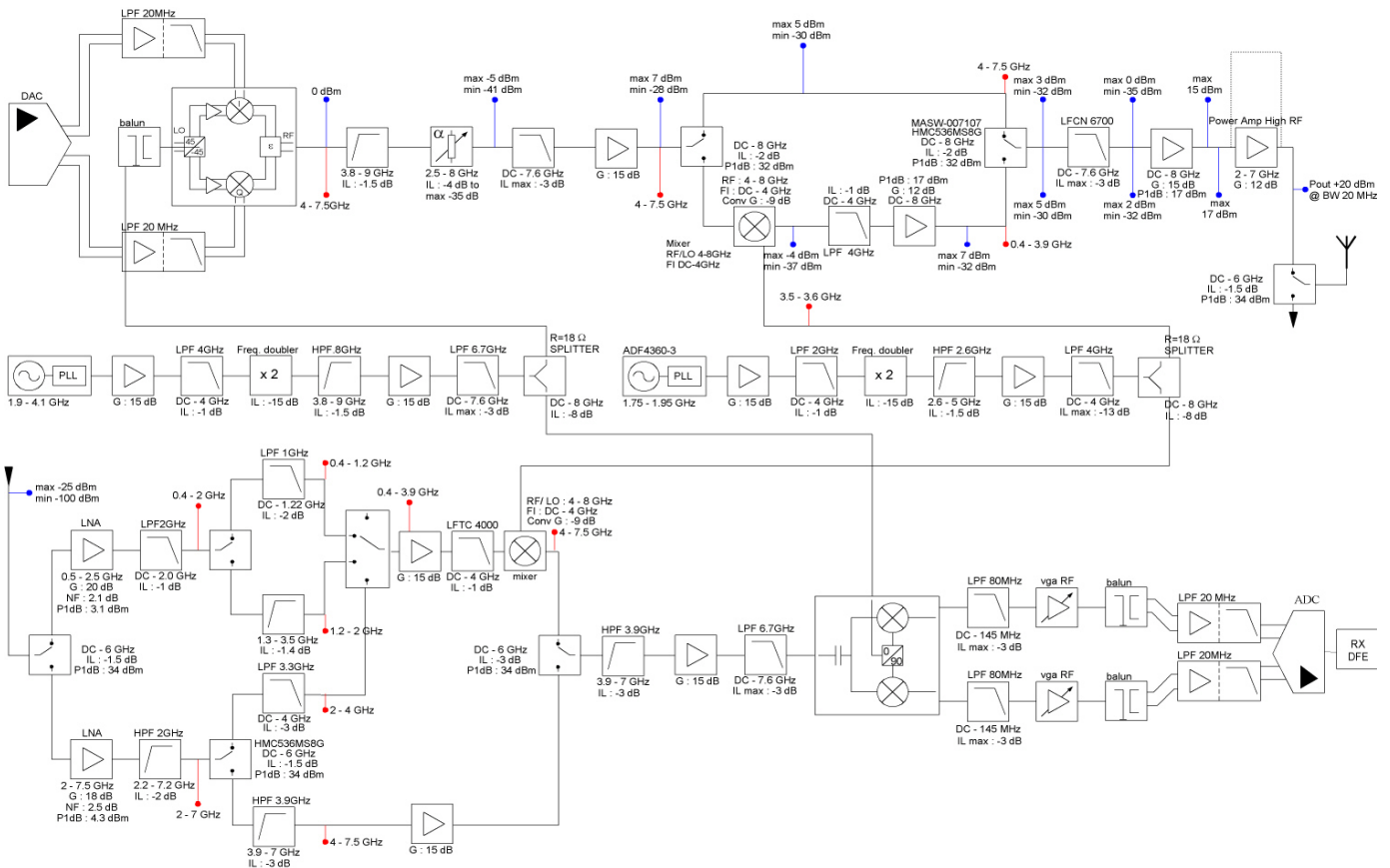
- It will allow communicating at the same time in different bands and different waveforms.
- The board by default is Time Division Duplex based. It means that we use the same frequency band for Rx and Tx on one chain, and that a switch is used at the front end. In other words, one can only either transmit or receive. In order to implement a FDD system, one has to use 2 chains, one for the uplink, and one for the downlink.

### B. Local oscillators

Since we address a very wide frequency band, the local oscillator generation is a key feature of the RF transceiver. The LO generation is based on a wide band frequency synthesizer (1.9 to 4.1 GHz) and a frequency doubler. Hence the LO range is from 3.8 GHz to 8.2 GHz. The drawback of this solution is that the frequency step is quite large, but it can be compensated digitally on the base band signals.

### C. Transmitter section

The base band signal (zero IF) is in I/Q format, and fed to a quasi direct modulator. The chosen component allows one to generate a signal directly from base band to a frequency range from 4 to 8 GHz. The modulated signal is then filtered and amplified. Afterwards, a switch is used to separate low and high frequencies, if the signal shall be transmitted between 4 to 7.5 GHz.



**Figure 3: Proposed RF Prototype Architecture**

### D. Receiver section

This part is certainly the most difficult one, due to the very wide addressed frequency band (from 400 MHz to 7.5 GHz). LNA is a critical part, since the performance of existing wide band LNAs are not yet adequate (relatively high noise figure, non constant gain vs. frequency). Hence, the

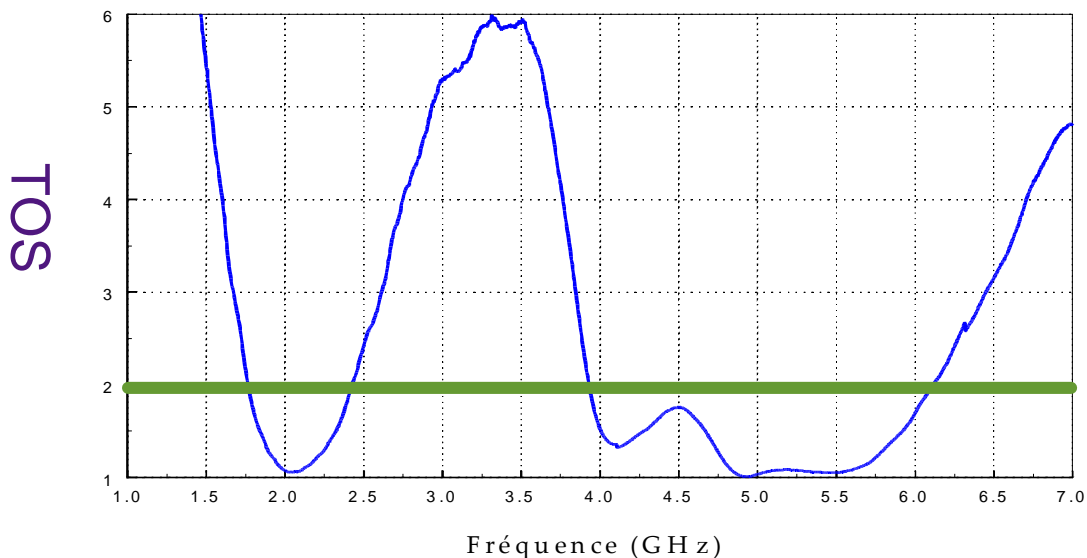
overall frequency band is divided (by a switch) into 2 sub-bands, one from 400 MHz to 2 GHz, and another one from 2 GHz to 7.5 GHz. After the LNA stage, the signal is filtered and the frequency bands are again spitted in 2 sub-bands. It gives us 4 sub-bands:

- 400 MHz to 1.2 GHz
- 1.2 GHz to 2 GHz
- 2 GHz to 4 GHz
- 4 GHz to 7.5 GHz.

This approach is used to decrease the amount of outer band interference (one has to keep in mind that basically all existing RATs are received at the antenna level, sometimes with huge level of signal). After this part, the architecture uses basically the same principle as for the transmitter section. We up convert the signal into a frequency range of 4 to 7.5 GHz and the signal is converted into base band thanks to an I/Q MMIC mixer. The base band signal is finally filtered and amplified.

### *E. Antenna*

Concerning the antenna part, the RF front end could be connected to a wideband antenna or a multiple band antenna. For example, antennas are available for both 2 and 5 GHz bands, as depicted in Figure 4. This figure illustrates the response of a multi band antenna designed by France Telecom R&D in the frame of the RNRT project Lao Tseu. Below the green line, the antenna is considered to be adapted to the frequency. One can see that this antenna supports both bands (2 and 5 GHz).



**Figure 4: Frequency Response of a Multi-Band Antenna**

## **2.3 Physical Layer**

In order to enable the proposed scenarios we need the following functionalities in the baseband:

- Ability to receive and transmit data through 2 RATs at the same time.
- Dynamic reconfiguration (for MIMO processing and discovery of a new RAT).

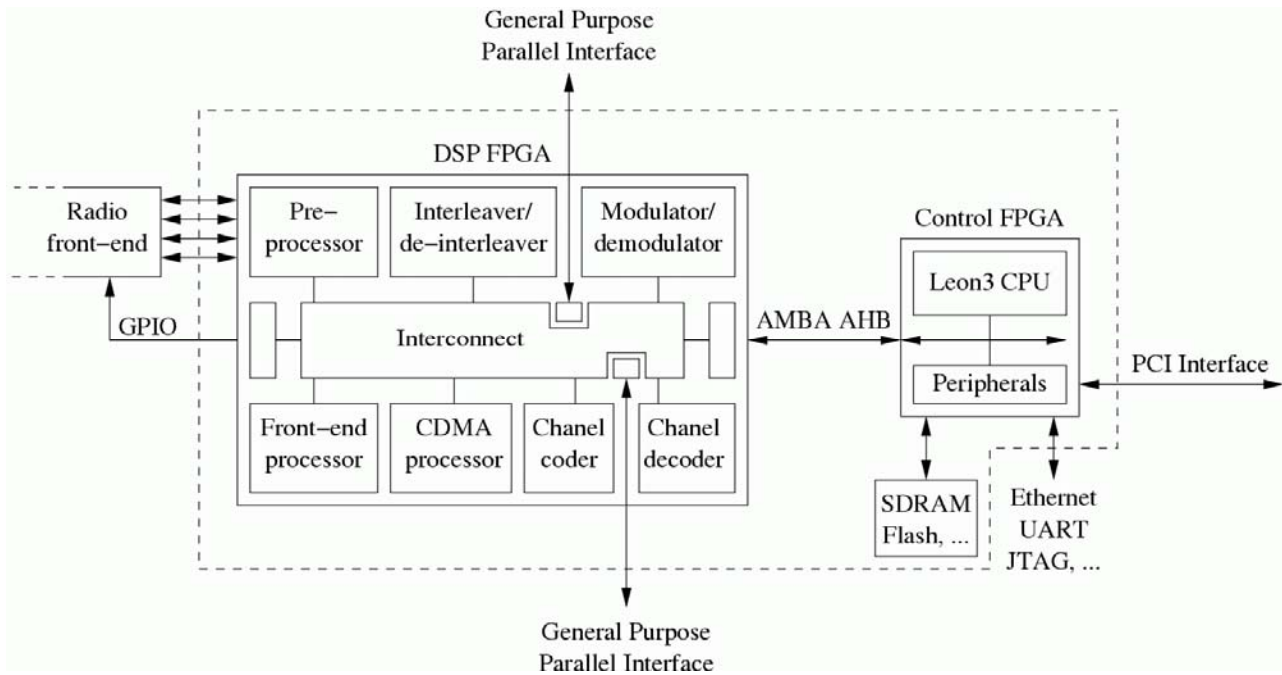
The purpose of IDROMel is not to have separate standard-specific components forming two distinct PHY layers (one for UMTS and one for WiMax), but to have the smallest set of highly reconfigurable computational units able to address these two standards. In order to achieve this goal several techniques will be used:

- **FPGA partial reconfiguration:**  
This technique gives the opportunity to reconfigure an unused part of a FPGA while an application is running on it. For example an IP used by WiMax MIMO decoding might be stopped and replaced by a UMTS-specific IP when a vertical handover is launched. This techniques works on Xilinx Virtex FPGAs.
- **A highly reconfigurable System-on-Chip (SoC):**  
The CEA/Leti FAUST chip will provide the MT with reconfigurable computing units addressing OFDM RATs. Its Network-on-Chip architecture will allow flexible routing of data among these units and from/to other partners' baseband contributions. The forthcoming FAUST2 chip will improve those characteristics by offering improved computing blocks and the ability to share its resources between 2 completely independent computing flows at the same time, thus addressing the challenge of having 2 simultaneously working RATs on the mobile terminal. When available it will replace the FAUST1 chip in the prototyping equipment.
- **Highly parameterised digital signal processing engines with common interface, in a dedicated FPGA:**  
(see next section)

### **2.3.1 ADAC/Baseband Subsystem**

The ADAC/BB subsystem will use two FPGAs and will offer most of the required functionalities. The Figure 5 depicts the subsystem architecture. One of the two FPGAs implements the high level control functions and the interface with the host system. Its main component is Leon3, a Sparc V8 micro-controller ( [3]). The system offers several different external interfaces:

- PCI/Cardbus
- Ethernet
- Serial
- GPIO
- General purpose parallel interfaces (e.g. FAUST interfaces)
- JTAG, ...



**Figure 5: ADAC/BB Board Architecture**

An external SDRAM and a flash memory are attached to the Leon3 CPU. The flash memory is used to configure the FPGAs and also as a standard non-volatile memory space. The software components can thus be stored in the flash (standalone mode) or downloaded from one of the external interfaces.

The control FPGA is connected to the other, larger, DSP FPGA through an AMBA AHB bus (AMBA/AHB is the native bus of the Leon processor).

The DSP FPGA contains 7 processing engines, a high performance VCI-based interconnect, a AMBA to VCI bridge and a GPIO module. The 7 processing engines are:

- A pre-processor: interface with the radio front-end, signal conditioning, etc.
- A front-end processor: time-frequency conversions, dot products, component-wise products, energy, max, argmax, phase correction, etc.
- A CDMA processor: CDMA related transforms.
- An interleaver/de-interleaver: all the interleaving functions but internal turbo-code interleavers.
- A modulator/demodulator: BPSK, QPSK, QAM16, QAM64.
- A channel coder: block codes, convolutional codes, turbo-codes.
- A channel decoder: block codes, convolutional codes, turbo-codes.

Each one of these IP blocks is highly parameterised and supports a wide range of different algorithms. They all embed a DMA engine and share the same generic interface with the interconnect. The local memories of the IP blocks are mapped in the global memory of the system. Some of the processors also embed a small 8-bits micro-controller which can be used to control long sequences of complex processing.

The DSP FPGA also interfaces with the CEA FAUST devices through two general purpose parallel interfaces and two bridges between the VCI-based interconnect and the FAUST NoC. These

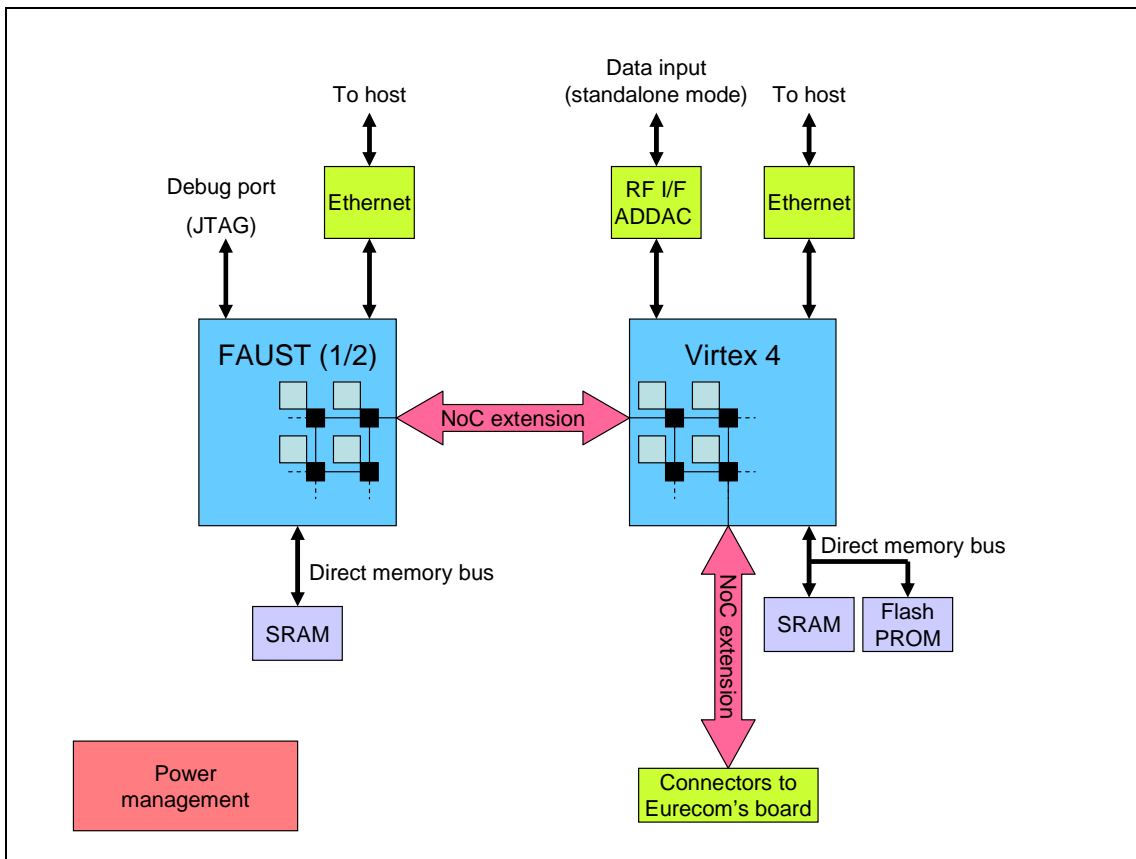
interfaces can also be used to communicate with other kinds of daughter boards.

### 2.3.2 FAUST board

In order to facilitate parallel developments and to minimize the risks, the FAUST chip will be integrated in a separate board. This board will plug in the the ADAC/BB board in the IDROMel demonstrator. However it will be able to work in a standalone mode, which gives more experimentation flexibility and eases the debugging process.

Figure 6 presents the architecture of the daughter board. It is built around a FAUST chip and a Virtex 4 FPGA from Xilinx. Both provide Telecom-dedicated IPs that exchange data through a Network-on-Chip. This NoC will be extended outside the ICs on the board and possibly to the DSP FPGA on the ADAC/BB board through its general purpose parallel interface. Thus the NoC benefits — in terms of bandwidth, dataflow distributed control and reconfigurability — will be shared amongst these components. Besides, the Virtex 4 FPGA makes it possible to exploit the FPGA partial reconfiguration technique.

The daughter board will also provide classical experiments and debug facilities, such as JTAG, Ethernet links, etc.



**Figure 6: FAUST Daughter Board Architecture**

## 2.4 Protocol Stacks

In the IDROMel project, we will mainly develop a layer 2, higher layers have been developed in previous projects and will be reused.

The UMTS MAC specification aims for the primary goals of UMTS LTE, even though the specifications for the latter are in the early stages of their development. The primary similarity with

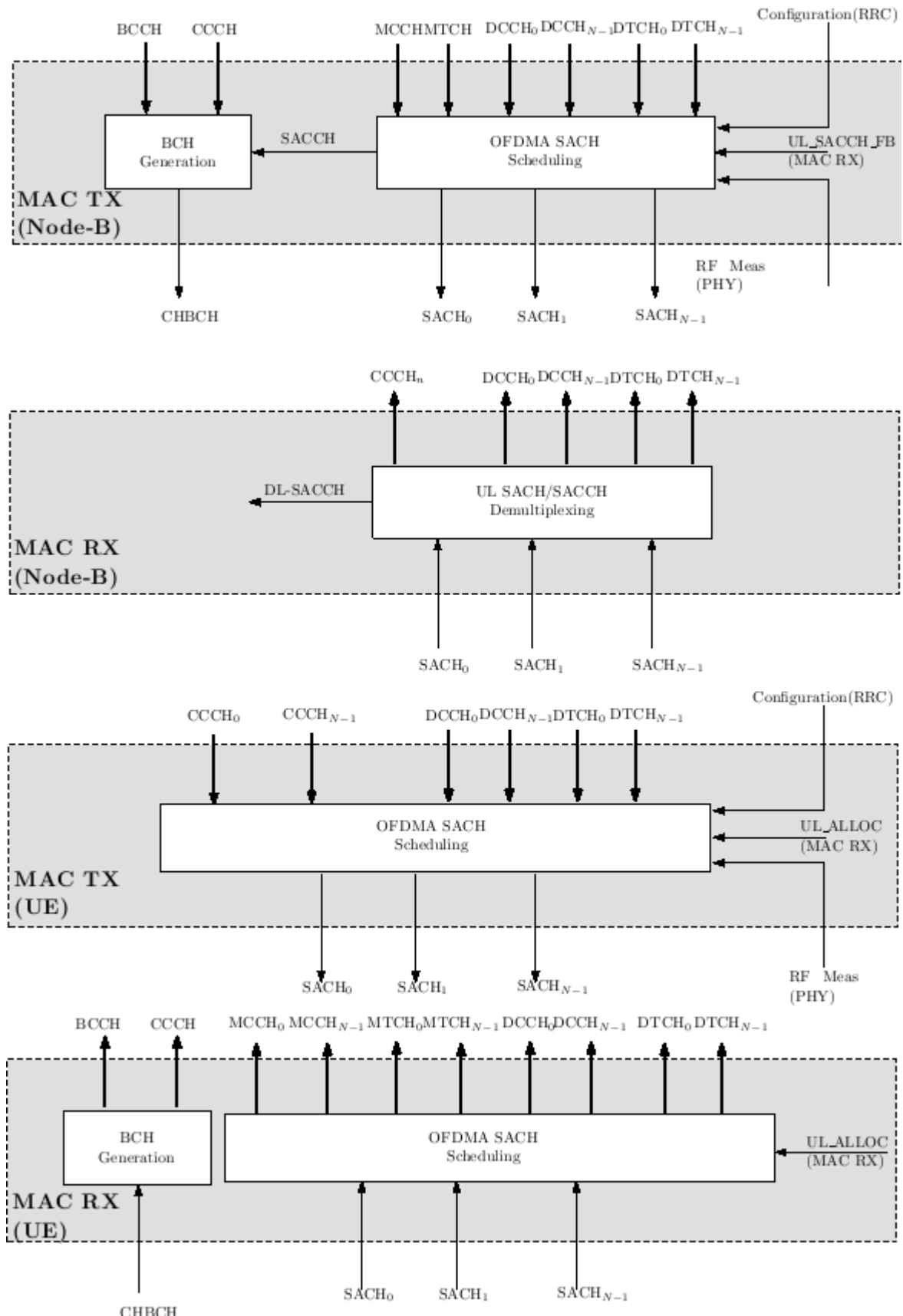
earlier versions of the 3GPP MAC layer (both TDD and FDD) is that the HSDPA transport channels are maintained, although they are adapted for OFDMA physical channels as opposed to WCDMA OVSF signaling. Dedicated channels, aside from CHBCH (BCH in 3GPP), are no longer present. All traffic, both UL and DL, pass through shared channels similar to HS-DSCH (DL). These are referred to as Scheduled-Access Channels (SACH). The different signaling channels, along with their comparable counterparts in HSDPA (Release 6) are described in [SACH Scheduling](#) ( [4]).

An ad-hoc MAC layer, derived from WiMAX 802.16 (and WiFi 802.11), is also used to achieve the demonstration of the scenarios specified in deliverable D1.1. The corresponding stack is shown in § 2.4.2, in particular logical channels are described.

These two stacks shall be used to demonstrate the IDROMEL scenarios. In particular, a reconfiguration from one to the other is required by scenario C.

#### **2.4.1 UMTS MAC-Layer Architecture**

The following figure shows the architecture of the MAC layer for both basestations (Node-B) and terminals (UE).



**Figure 7: Openair Cellular MAC**

The MAC layer is responsible for scheduling control plane and user traffic on the physical OFDMA resources. On transmission, the inputs to the MAC layer are connected to data queues originating in the RLC layer which form the set of logical channels. The control plane traffic is represented by the following logical channels:

1. Broadcast Control Channel (BCCH): This resource is a low bit-rate control channel used by the cellular network (via the basestation) for broadcasting basic information to users the cell served by a particular basestation.
2. Common Control Channel (CCCH): This resource is a low bit-rate control channel used both by user terminals and the basestation during the attachment or association phase of a new user terminal. Requests to join the cell are made by the terminal and acknowledgements are given by the basestation.
3. Multicast Control Channel (MCCH): This resource is a low bit-rate control channel used by the cellular network (via the basestation) to dynamically dimension the resources of the Multicast traffic channels.
4. Dedicated Control Channel (DCCH): This is a resource used by either the basestation or user terminal to relay access-layer signaling information (link-layer return channels, RF measurement reporting, traffic measurement reporting, power control, etc.) to the correspondent node.

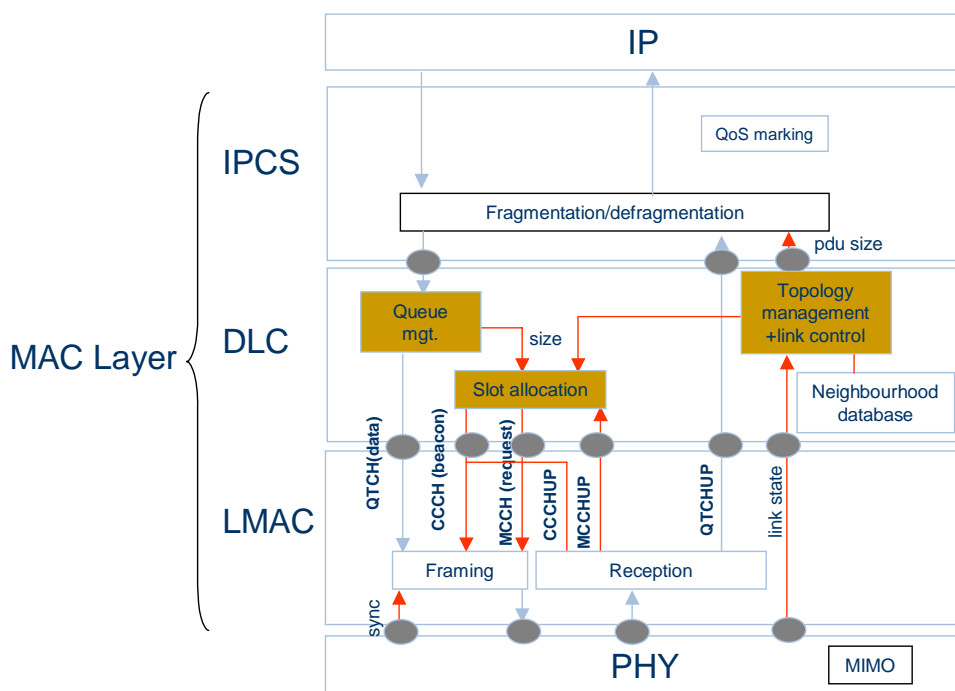
The user plane traffic is represented by the following logical channels:

1. Multicast Traffic Channel (MTCH): This resource is a variable bit-rate traffic channel used by the cellular network to relay multicast information to the groups of users in the cell served by the basestation.
2. Dedicated Traffic Channel (DTCH): This resource is a variable bit-rate traffic channel with negotiated QoS parameters used by the cellular network or user terminals to relay data traffic.

It is important to note that although dedicated resources are configured at the input of the MAC-layer, the physical resources allocated in the scheduling entities (with exception of the CHBCH) are dynamically allocated with the granularity of the mini-frame (nominally 2ms), and thus all physical resources are shared. Furthermore, in the case of TDD deployments, the portion of bandwidth allocated to uplink and downlink traffic is also dynamically adjusted at the granularity of the mini-frame.

#### **2.4.2 Ad-hoc MAC layer Architecture**

The following figure shows the MAC layer architecture of the nodes of the ad-hoc cluster:



**Figure 8: Ad-hoc MAC layer architecture**

The ad-hoc MAC layer is composed of IPCS (IP Convergence Sublayer), DLC (Data Link Control) and LMAC (Lower MAC) sub-layers. IPCS provides user data PDU to DLC sub-layer. LMAC sub-layer is responsible for managing the logical channels delivered by the DLC sub-layer. The logical channels defined at DLC/LMAC interface for the ad-hoc cluster are the following:

1. Cluster Control Channel (CCCH): This logical channel is used only when the node is Cluster head. It provides the LMAC with the distribution of the slots attributed to the different nodes in the cluster. This information is called "Beacon". A channel named CCCHUP, received from the Cluster Head at LMAC level, contains the same information for the nodes that are not Cluster Head.
2. Management and Control Channel (MCCH): This logical channel (coming from DLC sub-layer for all nodes) provides the LMAC with all DLC PDU that are required to operate the management services, as for example capacity requests. A channel named MCCHUP is received from the other nodes. MCCHUP contains the management information for these nodes, which allows the DLC sub-layer to perform its tasks, like the generation of the Beacon if the node is Cluster Head.
3. QoS Traffic Channel (QTCH): This logical channel corresponds to the set of destination-QoS queues that contain the data PDU to transmit. A channel named QTCHUP carries the data PDU received from the other nodes in the cluster.

LMAC sub-layer provides PDU to PHY layer. Slot synchronization is provided by PHY layer to MAC layer.

## 2.5 Reconfiguration Management

### 2.5.1 Protocol stack reconfiguration architecture

To demonstrate the scenarios specified in deliverable D1.1, an architecture shall be defined for

protocol stacks reconfiguration.

The scenario A, consisting in both protocol stacks (defined in § 2.4.1 and § 2.4.2) running separately on the same platform, requires that the protocol stacks can co-exist on the platform. The scenario B, consisting in an ad-hoc protocol stack running in MIMO mode on the platform, does not involve specific requirements regarding the protocol stack reconfiguration architecture. The scenario C requires a reconfiguration architecture to allow a soft handover between the two protocol stacks. Software radio modules management mechanisms shall be used to install/uninstall the protocol stacks: modules interfaces shall be specified and adapted to allow these operations, and the scheduling management of the handover shall be defined.

## **2.5.2 PHY/RF reconfiguration management**

We first try to identify some clusters of similar functions, which will use some similar reconfiguration processing. In a particular case, we identify 3 clusters. In our point of view this could be easily generalized and adapted to the elected scenario. Then we identify 3 types of reconfiguration needs. To perform in these conditions an efficient reconfiguration we proposed in [5] a hierarchical reconfiguration management approach, which could be easily adapted to the IDROMEL scenario.

### **2.5.2.1 Functional “Clustering” of transmitter baseband processing**

The definition of a common framework for an SDR system needs to start from a multi-standards analysis. But the functions are very different in an entire multi-standards baseband chain. So, as it is presented hereafter, we classify functions into 3 classes in order to specialize reconfigurable hardware to these classes (cf. Figure 9). It's easier to define common operators inside a class and to optimize these implementations in dedicated hardware resources (cf. Figure 9). Functions of a class which have the same structure are viewed as generic functions. So they are handled through few parameters to be adapted and to fit requirements of a standard. For example, the main parameters of the convolutional coding have been chosen for the wide variety of baseband signal processing represented. Effectively, GSM is a classical mono-carrier TDMA system, UMTS uses CDMA techniques, and finally 802.11g offers a multi-carrier mode (OFDM). Currently, we focus our attention on the transmission chain of the terminal because our main goal is to concentrate efforts on reconfiguration issues, and a multi-standards transmitter involves all conceptual challenges with less processing complexity than a receiver. First, we group the multi-standards baseband functions into 3 functional classes presented below. These classes group functions which have similarities of data type and architectural needs as shown in Table II below.

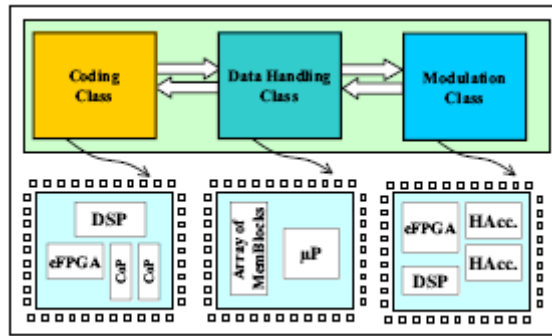
TABLE II  
3 STANDARDS BASEBAND PROCESSING FUNCTIONAL CLASSIFICATION OF  
THE TRANSMITTER

Class of Functions	GSM 900 (Uplink)	UMTS /FDD (Uplink)	802.11g (OFDM Mode)
<b>Coding</b>	-block Coding	-CRC Coding	-CRC Coding
	-Conv. Coding	-Conv. Coding	-Scrambling
	-Interleaving	-Turbo Coding	-Conv. Coding
	-Crypto	-(1 <sup>st</sup> , 2 <sup>nd</sup> ) Interleaving	-Interleaving -Interleaving
<b>Data Handling</b>	-Reordering	-Block Concat.	-Puncturing
	-Partitioning	-Block Segment.	-Phy Burst forming
	-Burst Building	-Frame Equal.	
	-Burst Multiplexing	-Frame Segment. -Rate Matching	
		-TrCh. Multiplex. -PhyCh Segment.	
<b>Modulation</b>	-Diff. Encoding	-Spreading	-Mapping
	-LGP Filtering	-Scrambling	-IFFT,
		-Mapping	-Cyclic Pref.
		-RRC Filtering	& Pilot Inser. -Symbol shaping

The **Coding Class** groups functions like cyclic coding, convolutional and turbo coding. These functions are generally based on a linear shift register structure and process one bit width data. Software implementations are possible but hardware ones are more efficient and consume fewer resources. Since data width is only one bit for this class of functions, software implementations are generally underoptimized because of using 16/32 bits processors. Furthermore hardware implementation is necessary to reach the expected high throughput performances of 54Mbps in 802.11g. But on the other hand, facing to the wide variety of coding schemes, flexibility of software is necessary. Consequently software implementation accelerated through efficient dedicated coprocessors corresponds to a suitable architecture for this class of processing functions.

**Data Handling Class** groups functions which manipulate data packets. Due to the functions nature (concatenation, segmentation, multiplexing, ...), data packet lengths are very different. These functions are exclusively dedicated to data transfers; it demands a lot of memory and this class of algorithms is largely control-oriented. So the high processing flexibility offered by processors is interesting. This is the reason why we associate arrays of memory blocks to this class of function. It is important here to pay attention to the architectural design to reduce the power consumption of arrays of memories by resizing it and switching-off unused memory blocks.

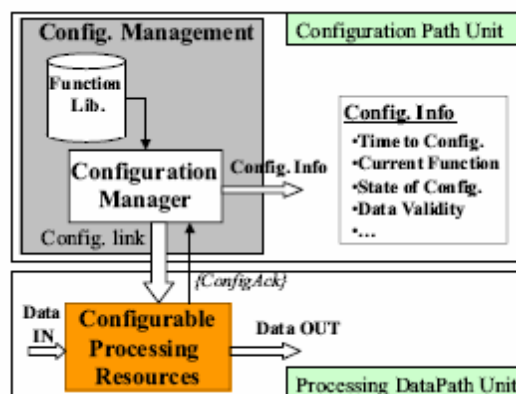
The **Modulation Class** corresponds to baseband functions which are just before transposition on frequency carrier. In this class, functions are computation-oriented and often process oversampled data, so high throughput is needed. Moreover, most functions handle data, up to 32 bits, which takes lots of resources. Consequently, some of these functions, like pulse shaping filtering will take advantage of an implementation in dedicated configurable hardware accelerators.



**Figure 9: Classification Demands of Computing Resources**

### 2.5.2.2 Configuration Management Considerations

Configuration management constitutes an important part in SDR systems and brings the reconfiguration facilities that are necessary to handle a growing number of standards. Standard switching corresponds to one type of application among others which require reconfiguration. Reconfiguration in Software Radio is largely discussed in literature [7] and as explained by Kountouris et al. [6] many reconfiguration needs appear in SDR. Reconfiguration is needed for bug fixing, processing reprogramming for performance enhancement or upgrade, mode switching, etc. Moreover, configuration management is closely related to the hardware resources as shown in the Figure 10. This constitutes an essential part of a reconfigurable system in order to handle new capabilities of configurable computing, and especially dynamic aspects of the reconfiguration. So the diversity in reconfiguration schemes, in addition to the heterogeneity of hardware within a platform, makes configuration management in SDR systems a complex issue. Requirements for each type of reconfiguration service are indeed different in terms of reconfiguration time or size of code to change and heterogeneous hardware implies to manage different types of code. For example, a configuration library could store cores IP for a reconfigurable hardware part of the system and software objects for embedded DSP or microprocessors. Thus, a configuration manager in a SDR system has to be able to handle multi-targets and multi-granularities configuration objects. Also, configuration management architecture must be an integrated part of the SDR System design.



**Figure 10: Functional Overview of Configuration Management**

### 2.5.2.3 The Hierarchical Approach of Configuration Control and Management

The great diversity of processing in a multi-standards application implies a large number of processing configurations to be managed. As the configuration management is complex and

represents an important part of the abstraction layer, we believe that our hierarchical approach of configuration control and management could simplify it. In the previous sections, the benefits of clustering the transmission chain have been presented and particularly to manage partial reconfiguration of an application. In addition to this, as the architecture is heterogeneous an entire configuration is made up of bit-level reconfigurable parts and words level reconfigurable parts, the proposal of a hierarchical view enables to manage multi-granularity of configurations. This functional model is illustrated by in Figure 11. The model is composed of 3 levels of hierarchy detailed below. A system architecture compliant to this functional model is composed with one *Configuration Manager* at level 1 as a host for the 3 clusters. There are several *Configuration Managers* at level 2 dedicated to each hardware components. At level 3, each *Configuration Manager* is responsible for a processing component.

#### 2.5.2.3.1 Hierarchical Level 1

At the first level, the processing path is split into 3 entities following the functional classification described in the first section of this paper. This first high level classification allows a more appropriate management of category-specific functions. The *Configuration Manager L1 (L1 CM)* works at the standard level as a host toward the underlying levels of management. This entity is in charge of choosing the functional units which will constitute the entire configuration of the baseband processing chain. At this level, generic functions are handled as generic black box components. Any hardware implementation is not yet considered in level 1. We focus our studies on the configuration management of physical layer, but we believe that (*L1 CM*) will deal with MAC layer to receive request of services from the user.

#### 2.5.2.3.2 Hierarchical Level 2

The generic functions selected in level 1 are specialized at level 2 in accordance with standards specifications. The set of attributes of each function is handled by the *Configuration Manager Unit L2 (L2 CMU)* to create each functional context of the entire processing chain. For example, the generic function “bit to symbol mapping” of the modulation class is set at level 2 to fit the selected standard. In the case of 802.11g standard this setting could be BPSK, QPSK, 16- QAM, 64-QAM constellation. At this level, in accordance to performance requirements of the function, the execution target is chosen. But the L2 CMUs don’t keep in charge the effective instantiation of the context onto the hardware. Here, functions are handled through few attributes to completely characterize them.

#### 2.5.2.3.3 Hierarchical Level 3

The processing datapath architecture of this third level will depend on the reconfigurable computing resources of the hardware architecture. In Figure 11, a part of the processing path is made up of several processing clusters. The complete processing path could be formed by different types of reconfigurable resources in accordance to the studies detailed in the first section. It could correspond to configurable accelerators, array of DSP blocks or a fine grain reconfigurable datapath. The main task of the *L3 CMUs* in the configuration path is to find available processing resources and configure it to enable the execution of the functional context created at the above level. Clock management of the resources is also included as an important feature of the configuration manager to reduce power consumption and to provide an efficient management of the multi-rate applications.

#### 2.5.2.3.4 Hierarchical approach advantages

This new high level hierarchical approach offers the opportunity to manage different granularities of the configuration demands. Depending on this granularity, management could be taken into

account directly at the right hierarchical level of the configuration path. This improves the overall performance of the SDR System and we illustrate this by the following reconfiguration cases. The reconfiguration requests for a complete standard switching will be managed differently compared with mode switching or little bug fixing.

- **Standard switching:** Standard switching is managed through the 3 hierarchical levels of the model as the transmitting chain will deeply change. For example, during a UMTS to GSM switching most of the parameters of generic functions are different and some function are standard-specific, then *L1 CM* will launch reconfiguration over most of the *L2 CMU* and *L3 CMU*.
- **Mode switching:** During a mode switching where just some parts of the transition chain have to be changed. For example in 802.11g OFDM, when a new data rate (from 18Mbit/s to 54Mbit/s) is requested, just some parts of the transition chain have to be changed as the mapping function (from QPSK to 64 QAM). Then few of *L2 CMU* and *L3 CMU* could contribute to the partial reconfiguration of the chain.
- **Little Bug Fixing:** When a problem of wrong code is locally well identified, a little reconfiguration is directly done by the concerned *L3 CMU*.

So, the hierarchical approach is appropriate for partial and multi-grain reconfiguration and will improve his control and management.

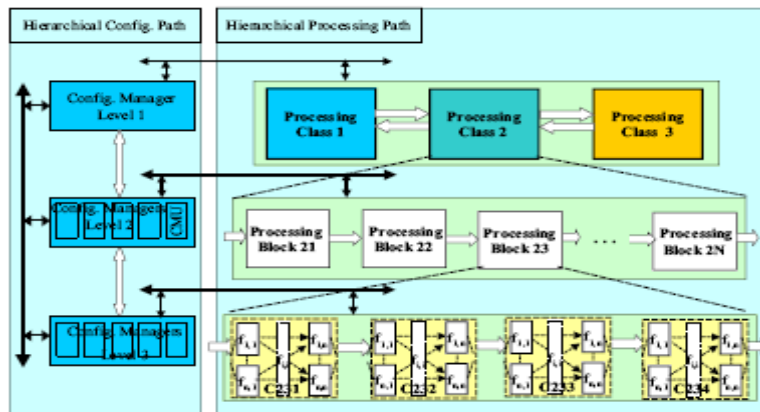


Figure 11: Hierarchical Model for SDR System Design

### 3 Conclusions

After a quick reminder of SDR key concepts — described in IDROMel deliverable #1.1 ( [1]) — that we plan to demonstrate in the frame of this project, we have presented the main requirements derived from the selected demonstration scenarios.

A foreseen global architecture of the demonstrator, fitted to the requirements, has then been presented, and a first vision of each one of its constituents has been detailed: RF front end, physical layer baseband processing boards, MAC and upper layers, as well as reconfiguration control and management considerations.

Thus this document constitutes a common base to detailed hardware and software specifications, which are the next step towards the development of the IDROMel demonstrator.

## 4 Reference

- [1] IDROMel Deliverable #1.1 “Rapports d’étude sur les tests et scénarios à implémenter”
- [2] E2R WP6 Deliverable 6.4 “Testbed Integration”
- [3] Gaisler Research website, <http://www.gaisler.com>
- [4] SACH Scheduling, in OpenAir website,  
[http://www.openairinterface.org/docs/html/group\\_mac\\_scheduling.html](http://www.openairinterface.org/docs/html/group_mac_scheduling.html)
- [5] *Jean-Philippe Delahaye, Jacques Palicot and Pierre Leray*, “A Hierarchical Modeling Approach in Software Defined Radio System Design”, SIPS conference, 2005
- [6] *A. Kountouris, Ch. Moy*, “Reconfiguration in Software Radio Systems”, 2nd Karlsruhe Workshop on Software Radios, Karlsruhe Germany, March 2002.
- [7] *M. Laddomada*, “Reconfiguration issue of future mobile software radio platforms,” *Commun. Mob. Comput.*, Ed.: John Wiley Sons, pp. 815– 826, 2002