Radio Virtual Machine (RVM)

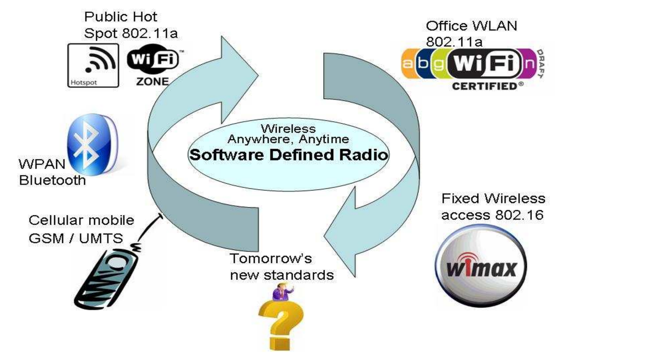
R. Ben Abdallah[1], T. Risset[1], A. Fraboulet[1], Y. Durand[2] and J. Martin[2]

*1 : CITI Laboratory - INSA Lyon, Bât. Claude Chappe,
6 avenue des Arts, 69621 Villeurbanne Cedex
{riadh.ben-abdallah, tanguy.risset, antoine.fraboulet}@insa-lyon.fr

*2 : CEA-LETI, MINATEC,
17 rue des Martyrs, F-38054 Grenoble
{Yves.durand, Jerome.martin}@cea.fr

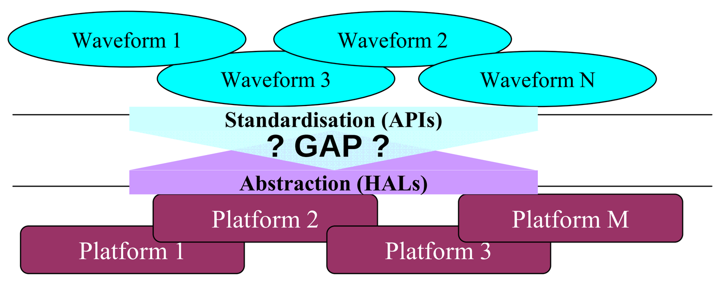
Software-Defined Radio (SDR) technologies

- A lots of promise
  - Lower costs
  - Faster time to market
  - Faster prototyping of new products
  - Easier update/upgrade of SDR products
  - Interoperability between SDR devices



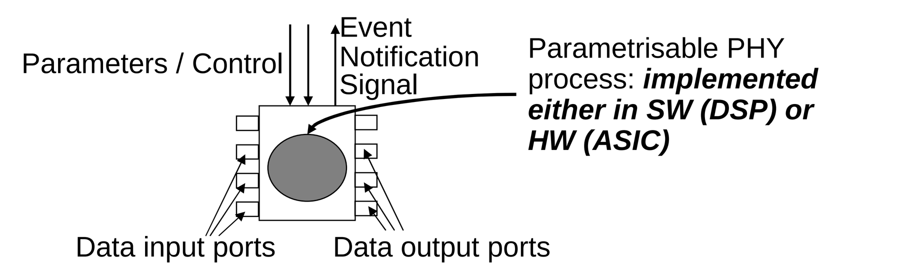
SDR supports present and future radio standards

- SDR implementation challenges
  - Reconfigurability ⇨ multi-standard radios
  - Portability ⇨ multi-platform radios
- Need for an adaptation layer
  - Expresses the convergence of PHY standards
  - Meets hard real time constraints
  - Hides Platform complexity
  - Extensible



Models and Proposals for the RVM

- New programming model for PHY Layer description
  - Abstract component view hiding implementation details



RVM manages functional components with well defined interfaces

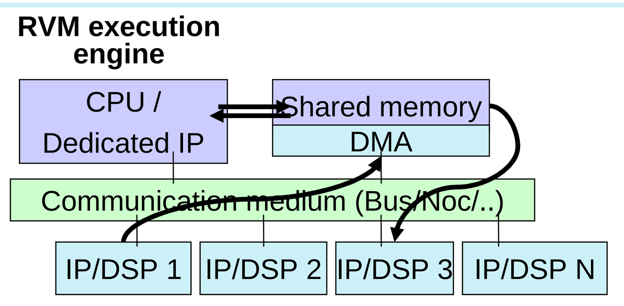
- PHY Layer Description Program (PLDP) expressed in a programming language dedicated to the description of physical layer protocol (sometimes called waveform language)

```
[...]
-- ressource allocation
dma1 = dma_engine.allocate()
phaseestim=phase_estimator.allocate()
rotor1 = rotor.allocate()
dma2 = dma_engine.allocate()
--*********************************--
--*** COARSE GRAIN DRIFT ESTIM ***--
-- VM connects allocated blocks
rvm.connect(RF.out, dma1.in)
rvm.connect(dma1.out, phaseestim.in)
rvm.connect(phaseestim.out, dma2.in)
```

```
-- block configuration
dma1.program("BEGIN \
{RECEIVE_AND_SEND 160} END;")
phaseestim.configure("STS", 160)
dma2.program("BEGIN  \
{RECEIVE 1; SEND_IT SIGTER;} END;")
-- VM waits for interrupt
(phase_drift,phase_amount)=rvm.wait(SIGTER)
--rotor configuation with phase drift
rotor.configure(phase_drift, phase_amount)
phaseestim.configure("LTS", 160)
[...]
```

PLDP program corresponding to the phase drift estimation

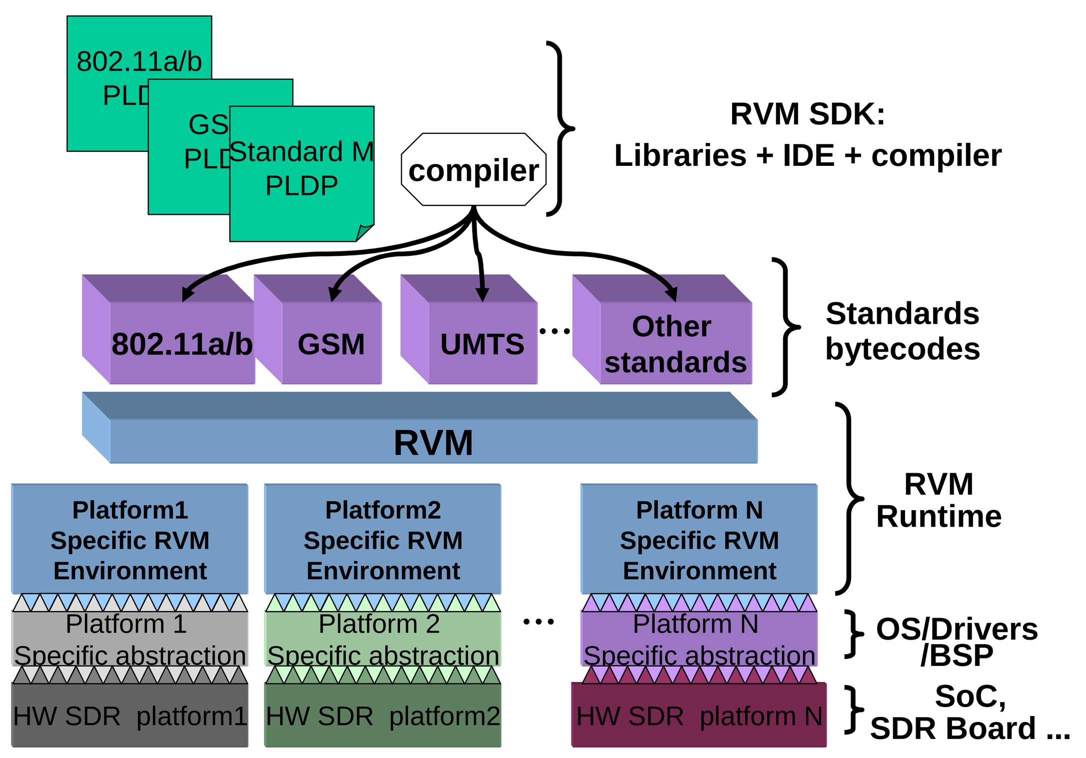
- Execution model
  - RVM is able to access the data-flow
  - Configurations could be sent :
    - Directly from the RVM
    - RVM sends requests to IPs to get their configurations from system shared memory



RVM is able to compute on the dataflow

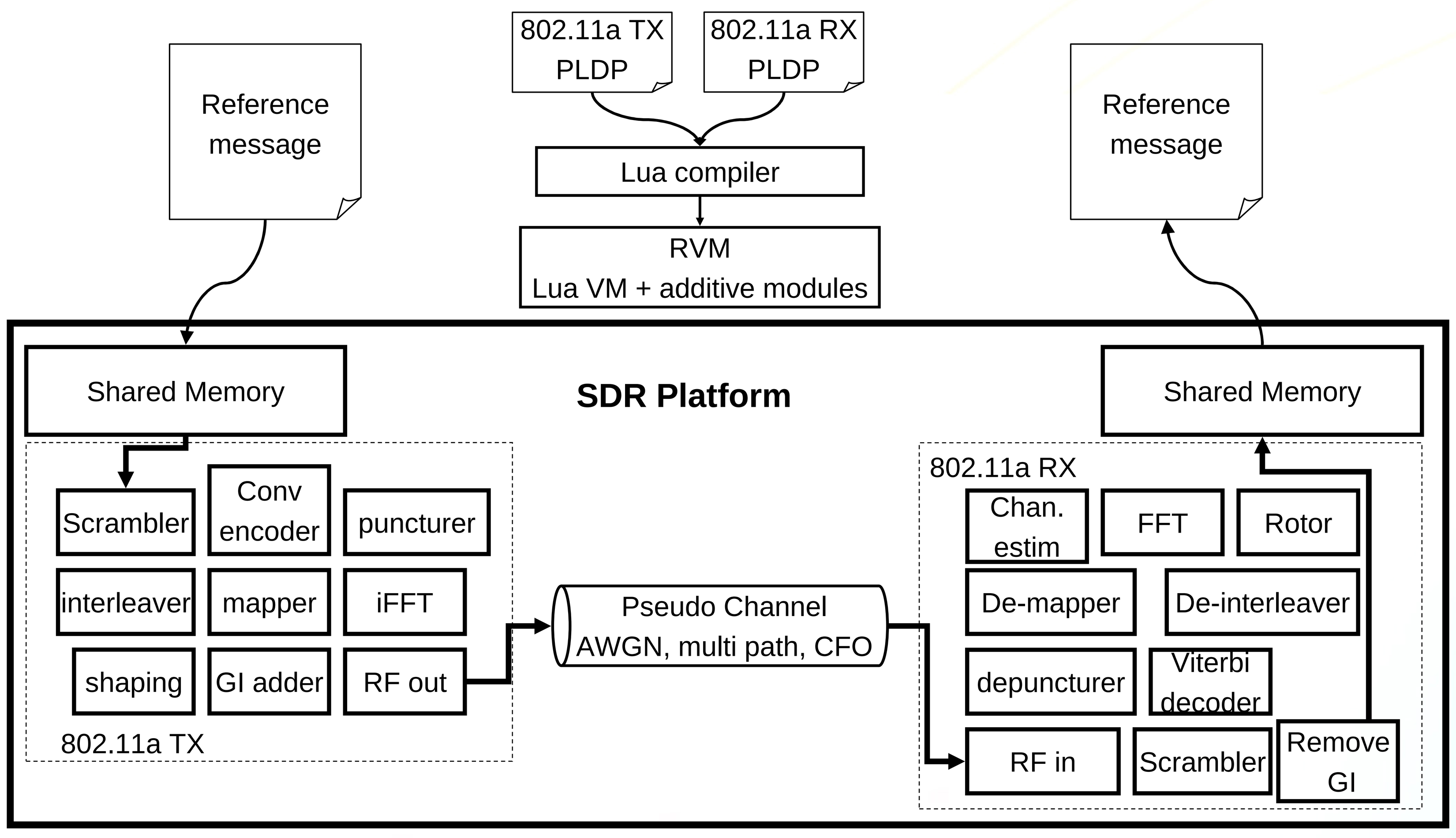
With RVM concept SDR meets its challenges

- RVM concept principles



- RVM advantages
  - Isolation of radio application from its execution platform
  - Dynamic software download
  - Maintenance and deployment of only one bytecode per platform
  - Implements processes with low computational requirements when no hardware support

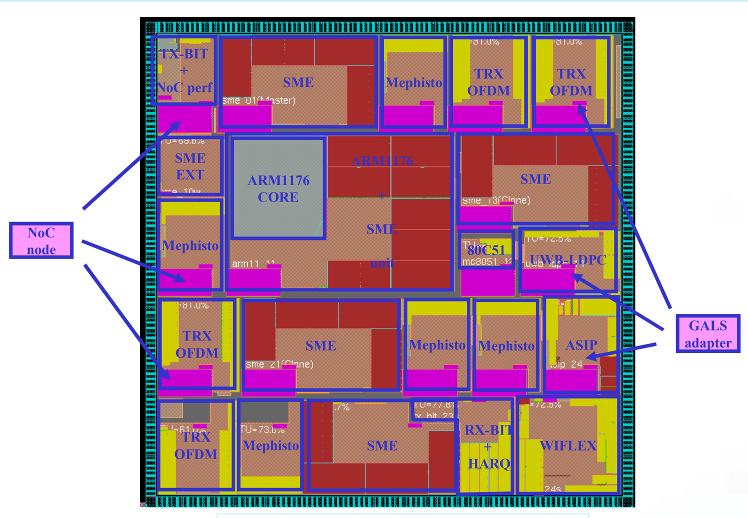
A RVM prototype running on PC: 802.11a PHY Layer On Lua RVM



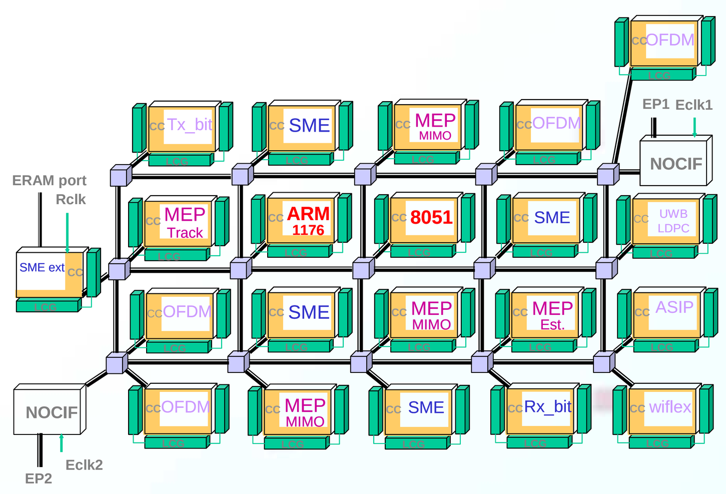
- Components are implemented in software on standard PC
- Components run in parallel threads
- 802.11a is functional (real time is not met of course)
- The "proof of concept" implementation is successful

Ongoing Work

- Real Time implementation of the RVM concept on the CEA-LETI MAGALI chip
  - Development of a MAGALI specific RVM environment upon F2 API (FAUST2/MAGALI HAL)
  - Experimenting RVM API using a lightweight Lua VM
  - Experimenting RVM with Squawk JVM(Java VM)
  - RVM runs on ARM1176 core with DBX (Direct Bytecode Execution) feature
- Evaluation of the RVM overhead
- Proposal of new paradigms and optimizations



MAGALI chip layout



MAGALI NoC

Leti www.leti.fr · MINATEC · INSTITUT CARNOT CEA LETI · INRIA · INSA LYON · CITI · cea