Model-Oriented Programming: A Simplified Programming Paradigm for Digital Architectures

Ludovic Apvrille,
ludovic.apvrille@telecom-paris.fr

ISAE-SUPAERO, Toulouse, France

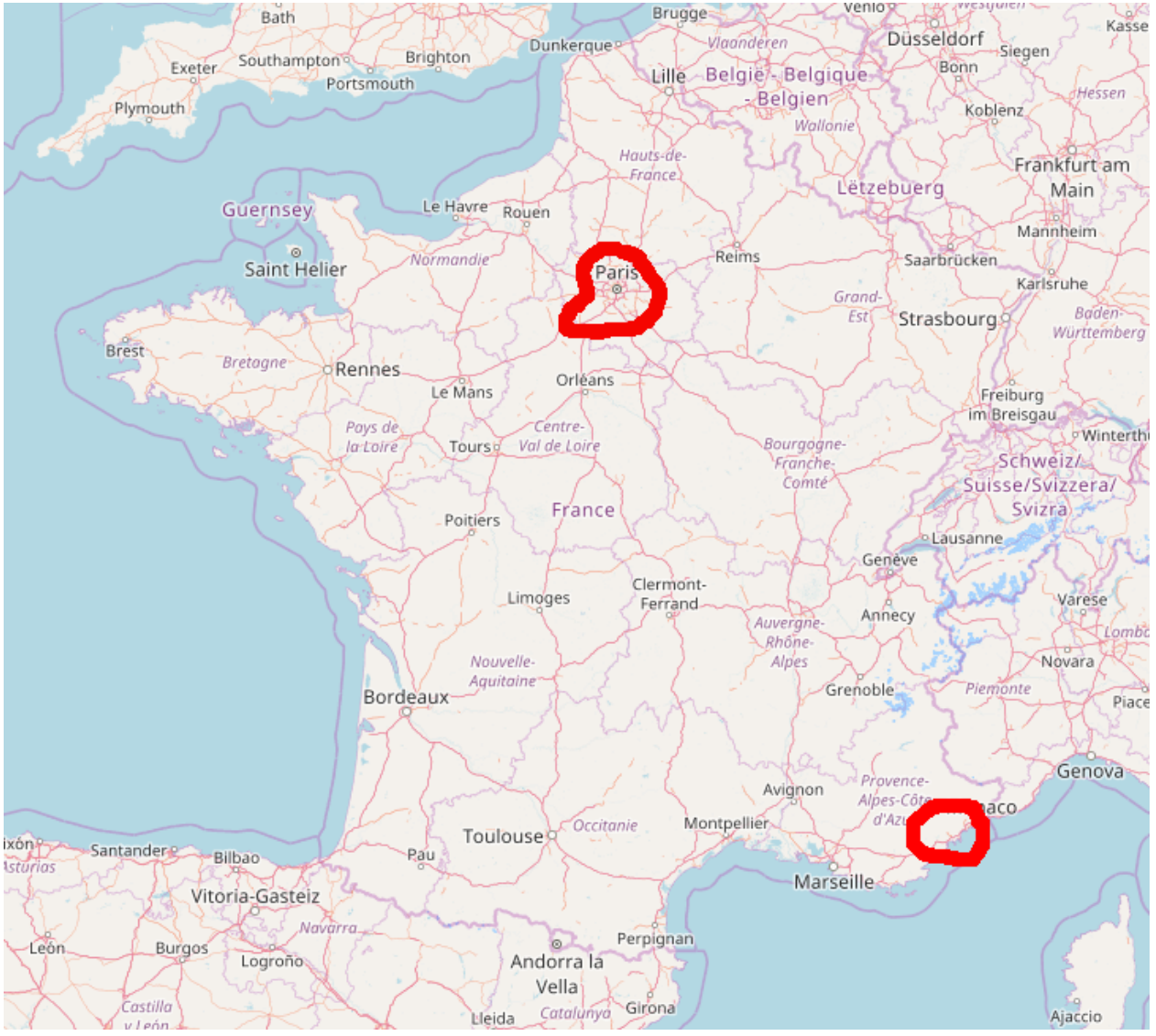
Outline

Outline

Members

- 5 Full-time researchers, 2 research engineers, 7 ph.D. students

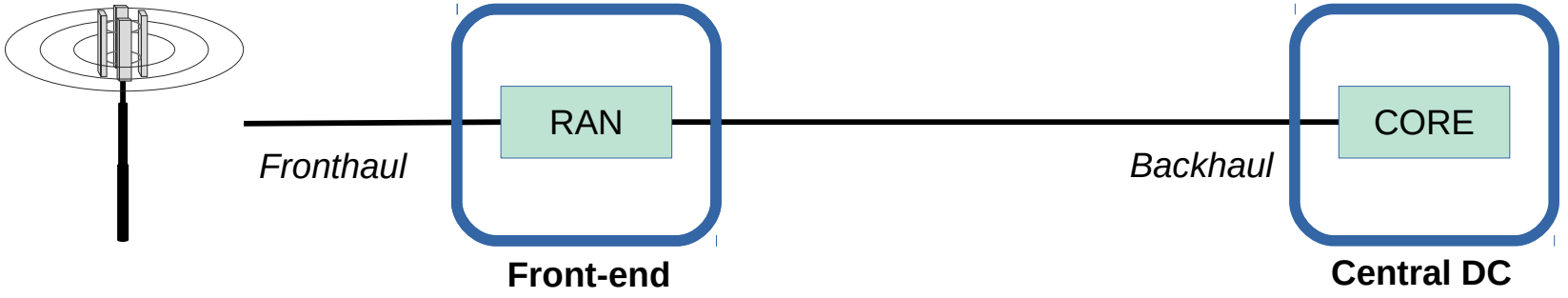
Paris and Sophia-Antipolis



Télécom Paris @ Sophia-Antipolis

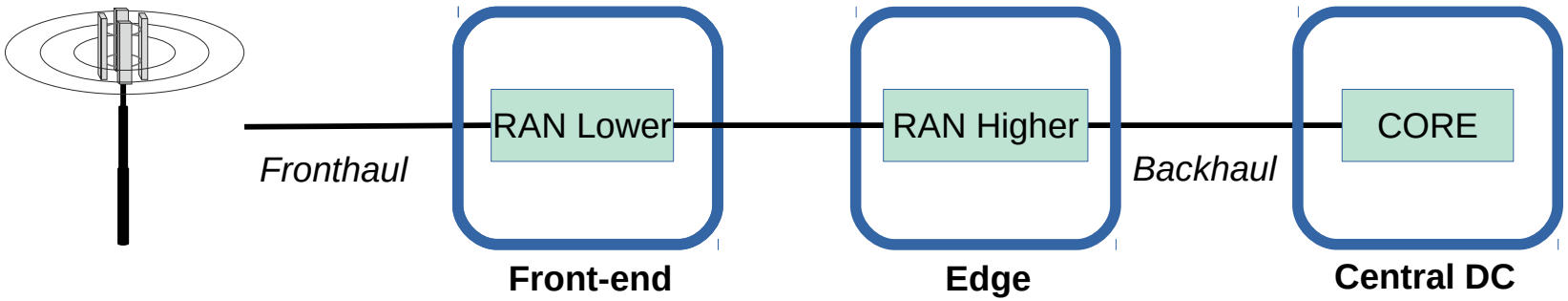
Outline

First Mobile Networks

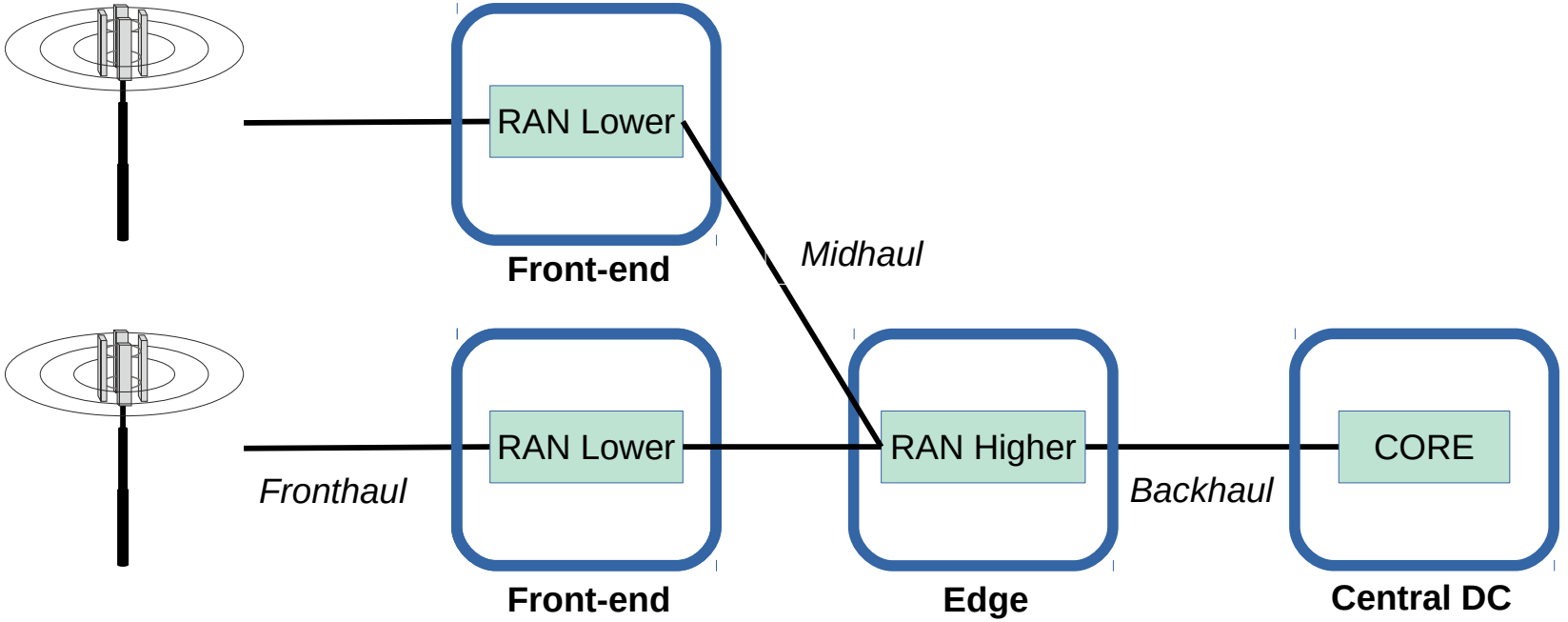


RAN: Radio Access Network
DC: Data Center

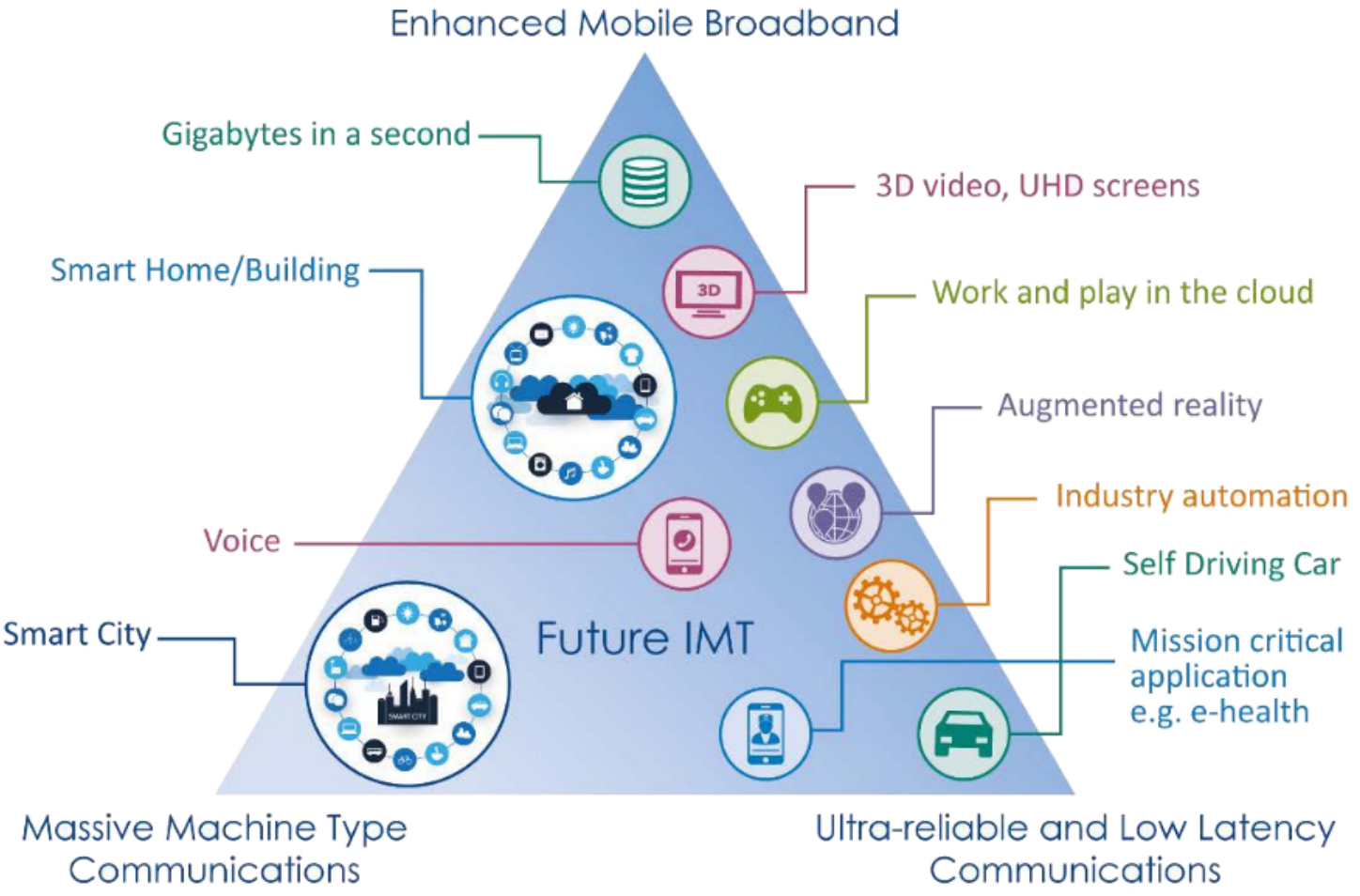
Evolutions


RAN Lower
RAN Higher
CORE
Fronthaul
Backhaul
Front-end
Edge
Central DC

We Are Here Today!

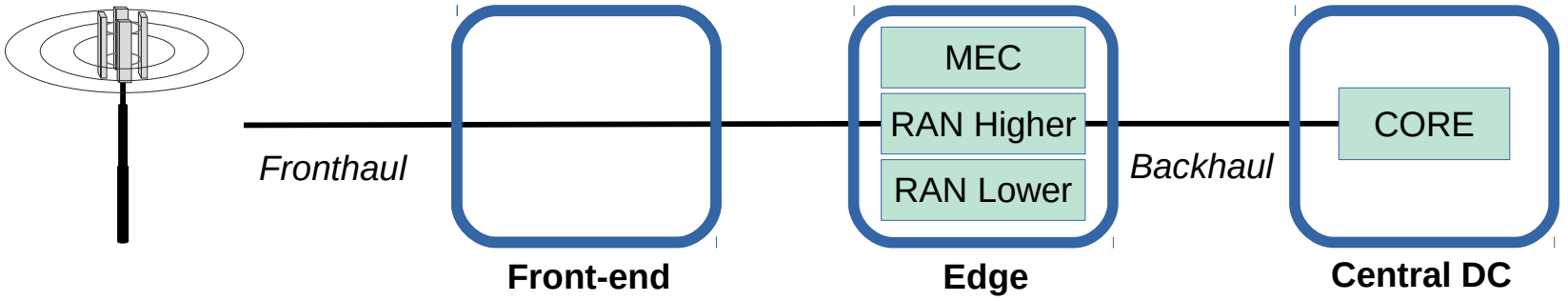


5G Constraints



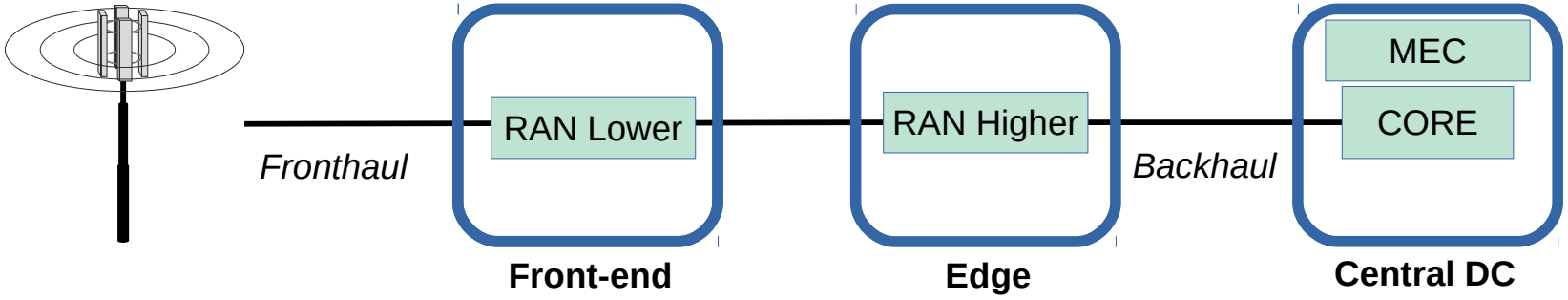
Source: ITU-R IMT 2020

Enhanced Mobile Broadband

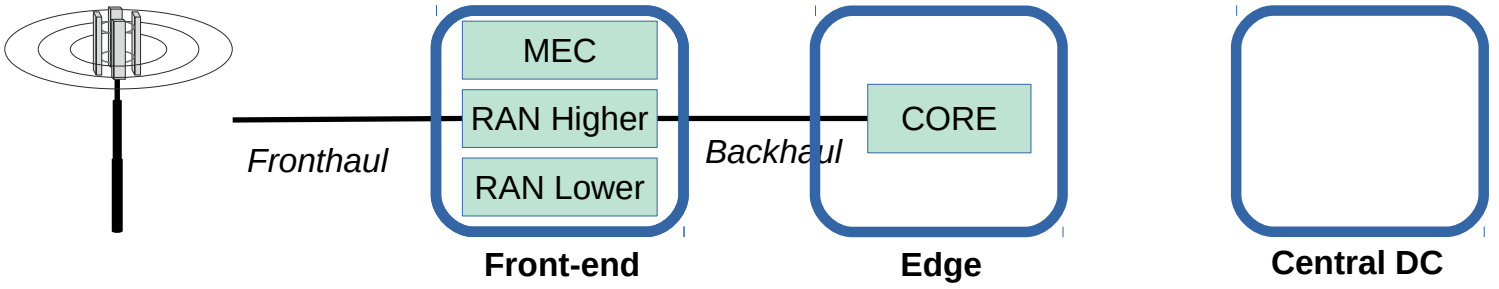


MEC: Mobile Edge Computing

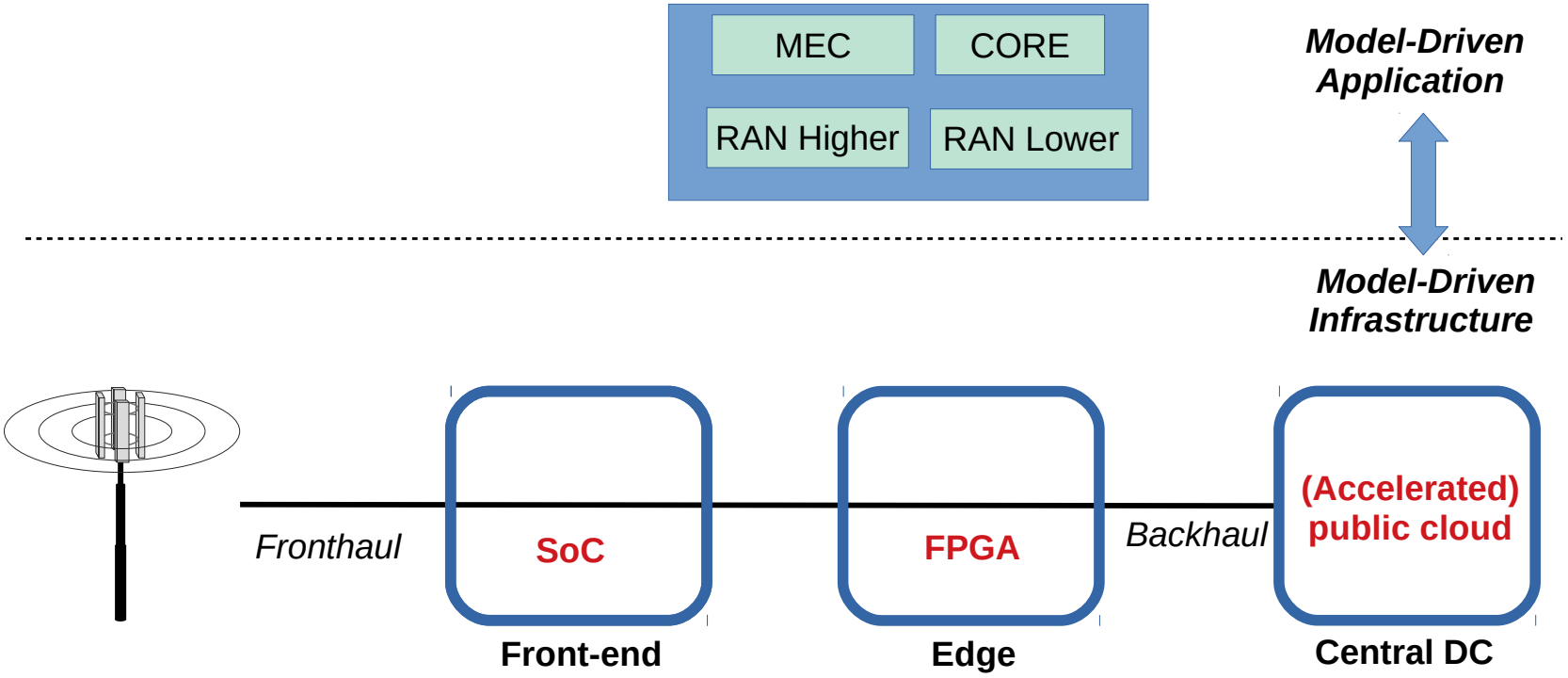
Internet of Things



Ultra Reliable and Low Latency



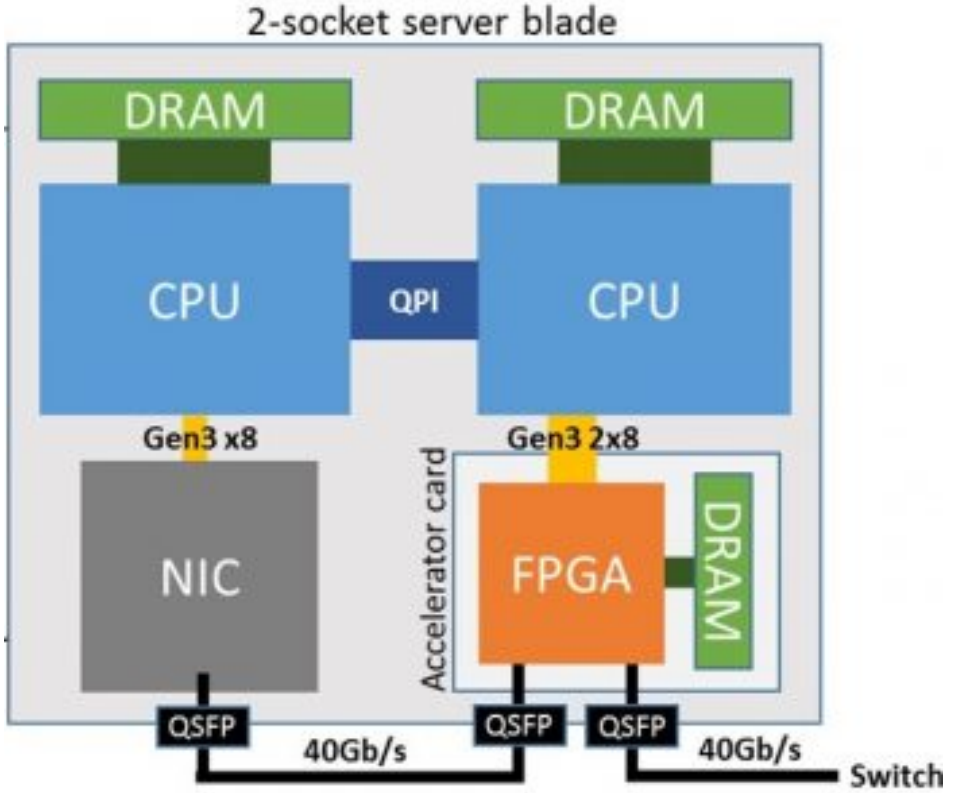
How to Select the "Best" System?



Accelerated Cloud

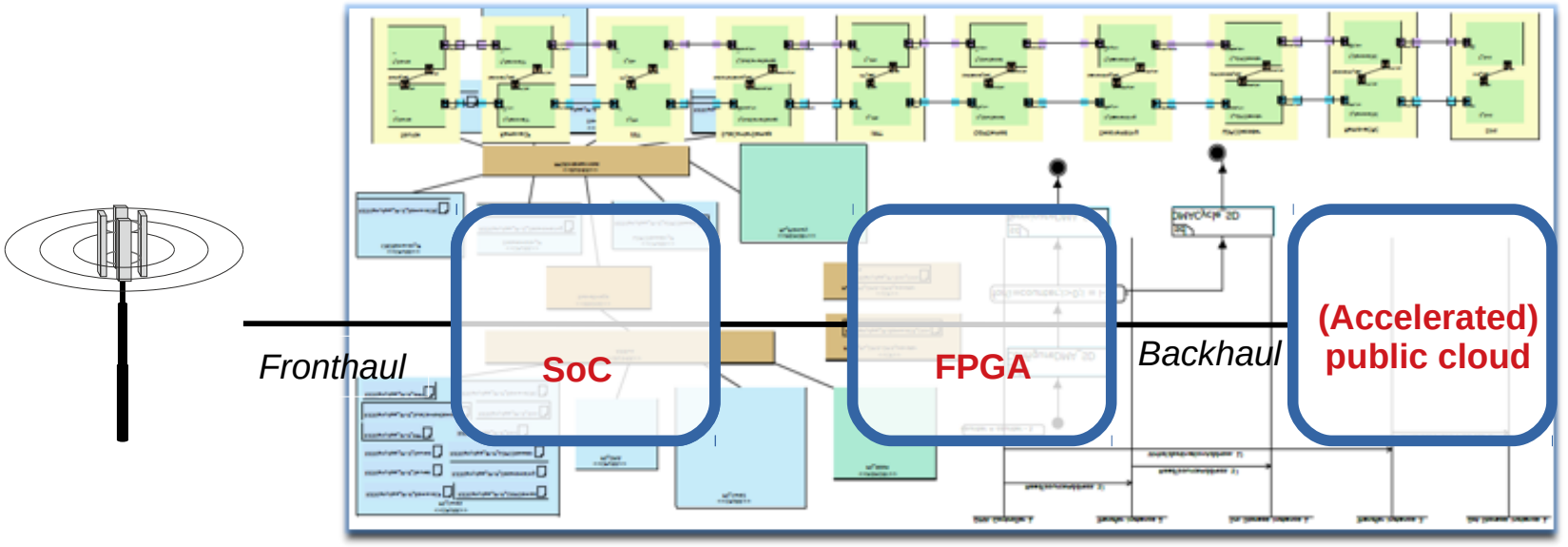
- Today: core network servers are used for non-real-time computations (bioinformatics, web search ranking) [1]
- Tomorrow: servers in cloud-based access networks will compute real-time operations (physical layer functions) [2]

Microsoft Catapult server



1] A. Caulfield et al: *A Cloud-Scale Acceleration Architecture*. In: IEEE Micro, pp. 1-13 (2016)
[2] A. Checko et al: *Cloud RAN for Mobile Networks*. A Technology Overview. In IEEE Comm. Surveys Tutorials, vol. 17, n. 1, pp. 405-426 (2015)

Towards Embedded Clouds



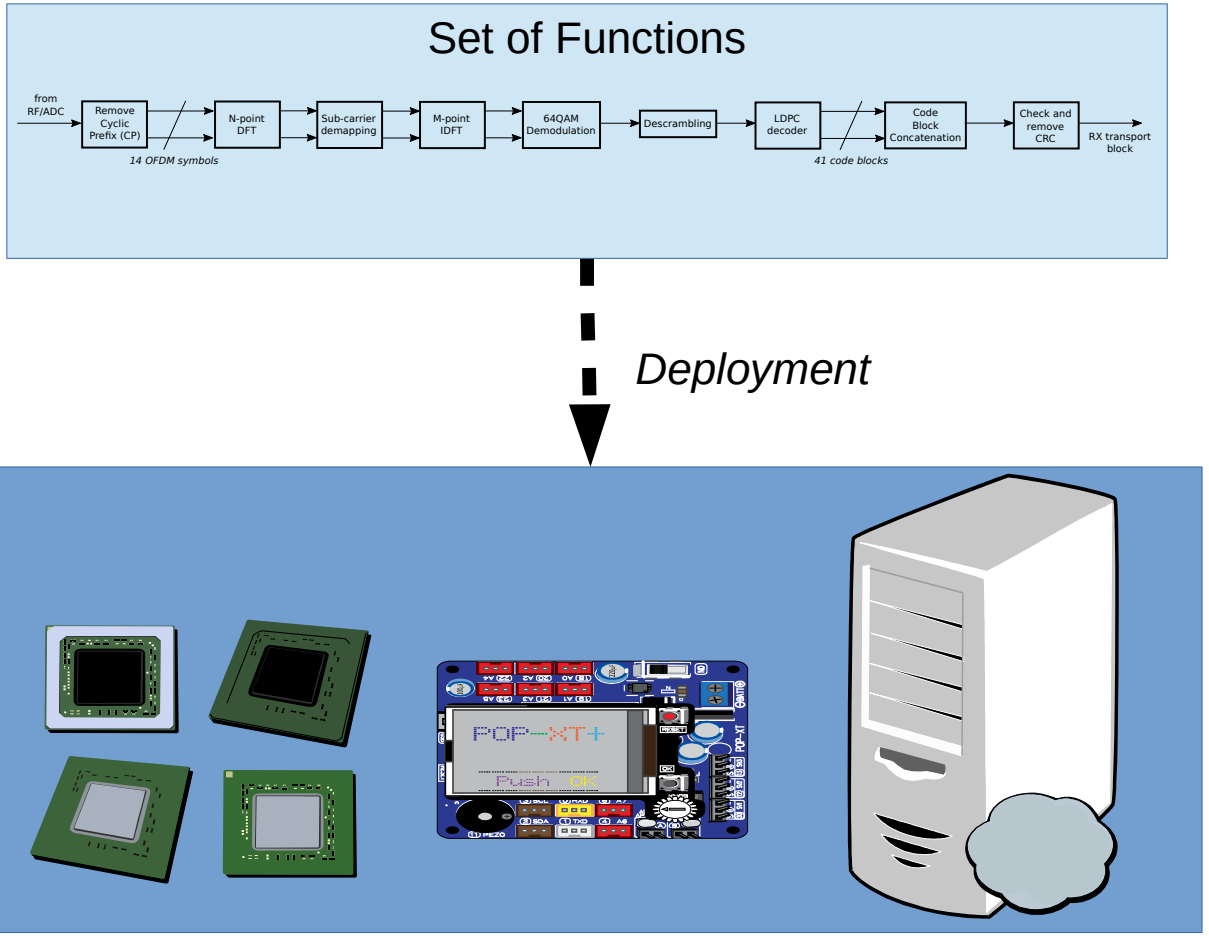
Federation of different systems (cloud) vs. a unique and giant embedded systems.

→ Use of the same design techniques for all resources

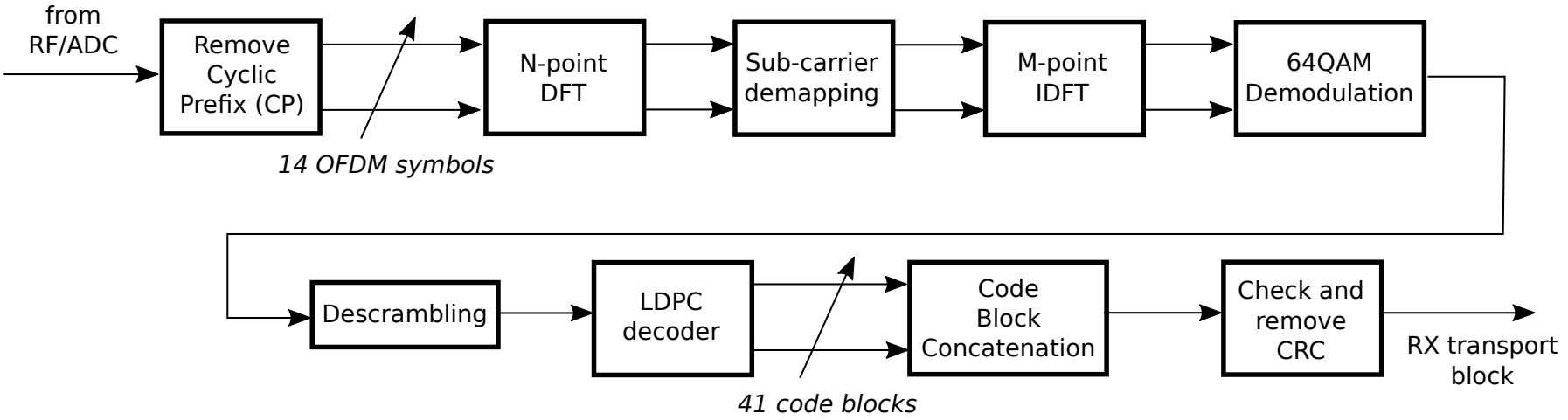
Outline

Objectives

Optimal deployment of signal-processing functions in future digital architectures



Example: 5G Physical Layer



Data-Dominated Applications

They need:

- To meet real-time constraints
  - e.g. end-to-end latency and have a predictable behavior
- To have a predictable behavior
  - Memory allocations
- Special hardware characteristics
  - Computation capabilities
  - Data transfer capabilities
  - DSPs, hardware accelerators, FPGAs, GPUs

Architecture Impact

Programming heterogeneous multi-processors is much more complex than parallel programming

- How to parallelize software?
  - Manually? Automatically?
- How to map parallel functionalities to execution units?
- Which execution unit, memory, communication protocol to select?
- How to interface hardwired accelerators?
  - FPGAs, GPUs, coprocessors

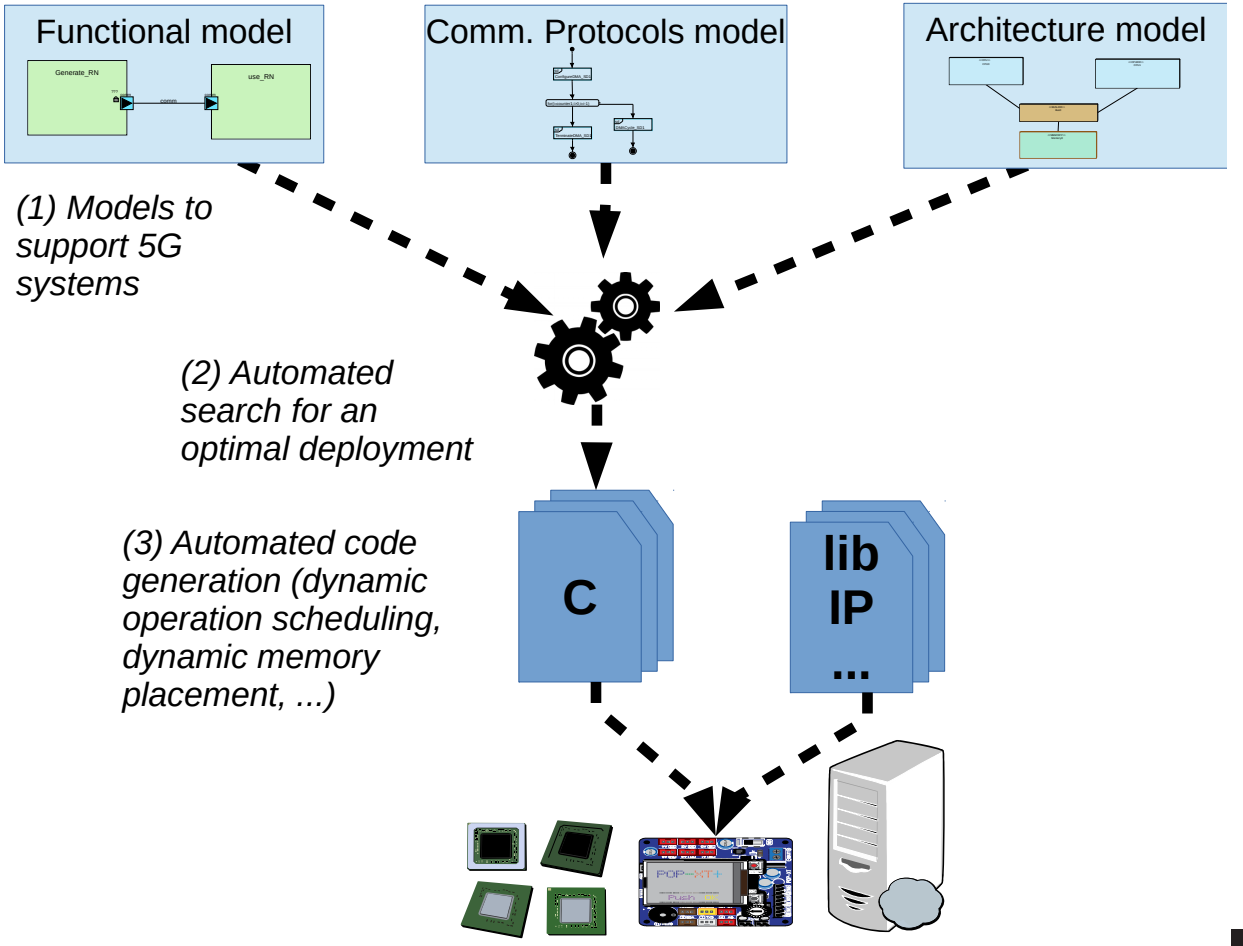
→ **Need for higher-level languages, tools and methodologies!**

Model-Based Software Engineering

"*An approach to systems and software development in which models play an essential role*" [Bran Selic]

Research Challenges

Model-based Approach



Partnership with Nokia Bell Labs

- *Model-Driven Programming of Digital Infrastructures*
- 3 PhDs, 1 research engineer
- 2017 $\rightarrow$ 2020
- Key persons in Nokia: Andrea Enrici, Laurent Roullet

NOKIA Bell Labs

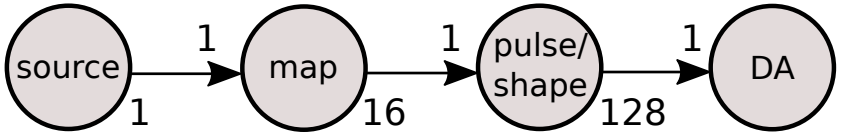
Outline

Models for Data-Dominated Systems

MoC: Applications as graphs: nodes/tasks, edges/buffers

Synchronous Data-Flow (SDF)

- Bounded FIFO buffers
- Non-blocking read, Non-blocking write
- Fixed production & consumption rates (e.g. no global variables)



The SDF graph for a pulse amplitude modulation system. Example from: P. Schaumont: A practical introduction to hardware/software co-design. Springer (2010)

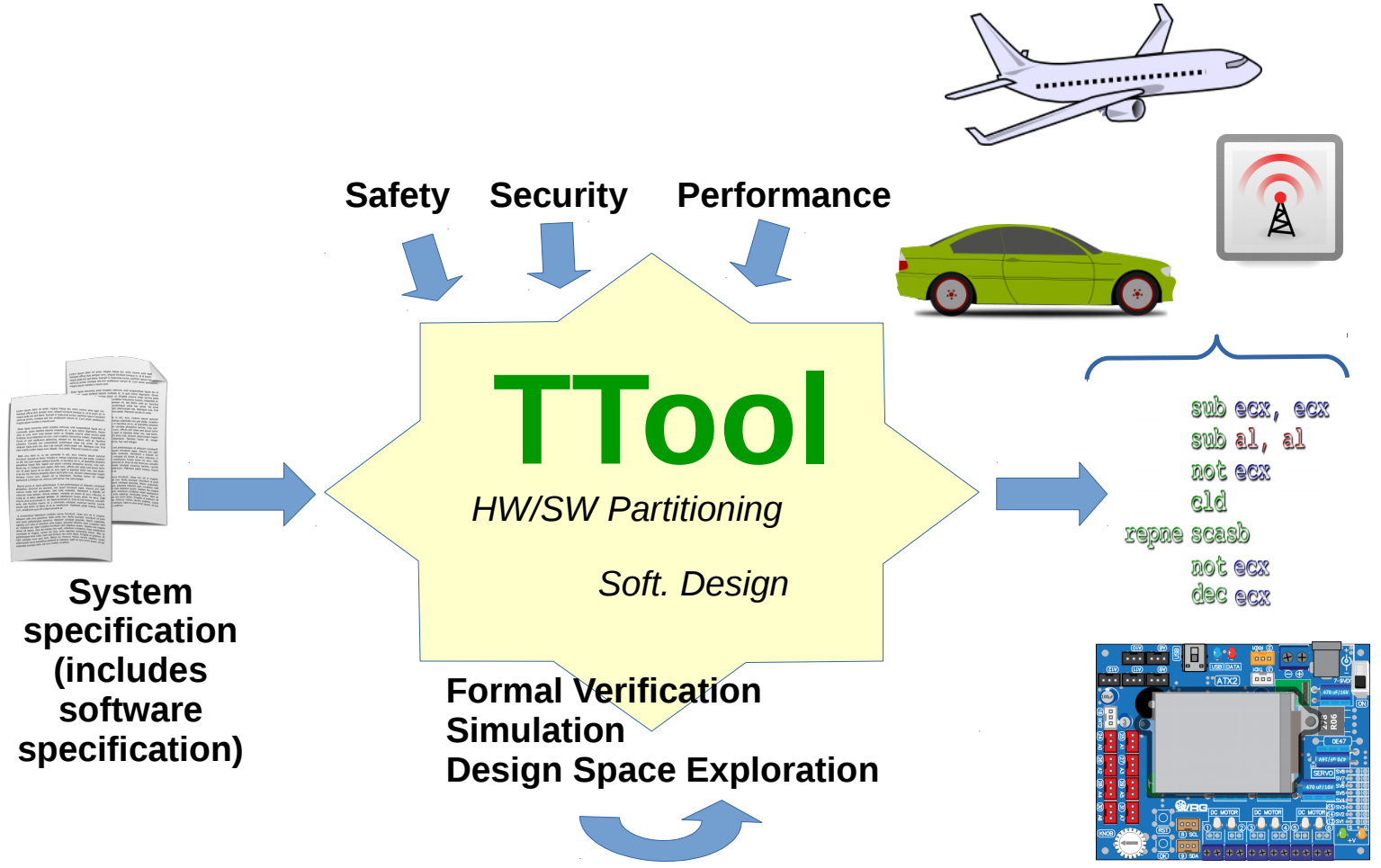
Process Networks (PN, Kahn PN)

- Unbounded FIFO buffers
- Blocking read, non-blocking write
- No global variables

UML/SysML

- TTool
- Block and activity diagrams

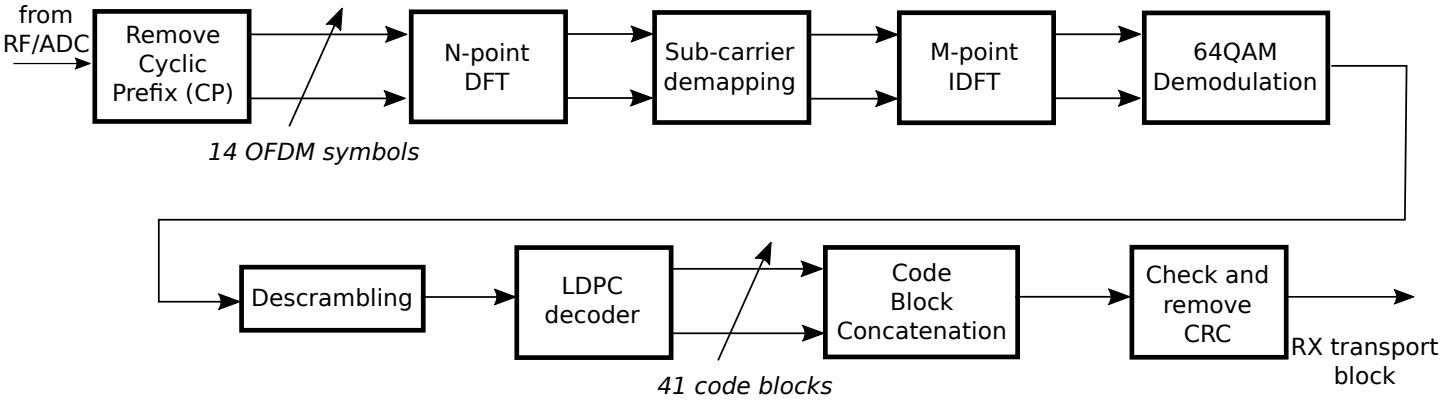
Goals of TTool

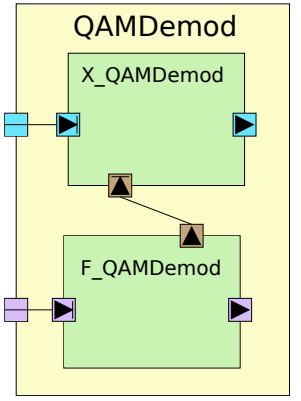


TTool: Key Features

- Model-Driven Engineering tool
- Free and Open-Source
  - Plug-in can be used to insert private/commercial features
- Easy to use
- **Focus on safety, security and performance**
- **Formal verification at the push of a button**
- **Code generation**

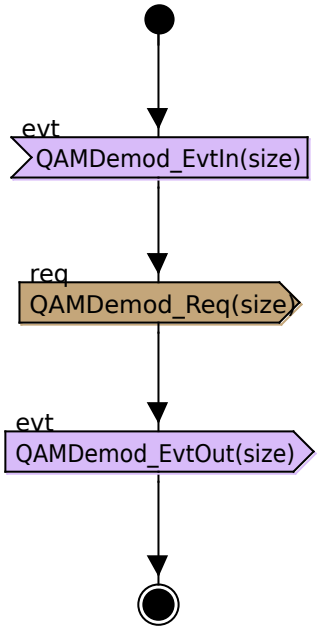
Example: 5G Physical Layer



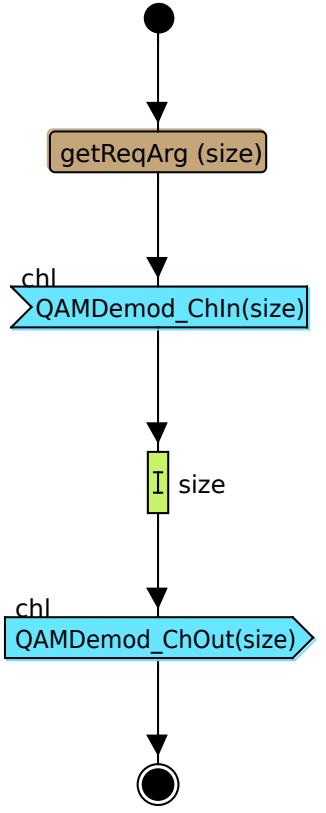


Example: Task Behavior

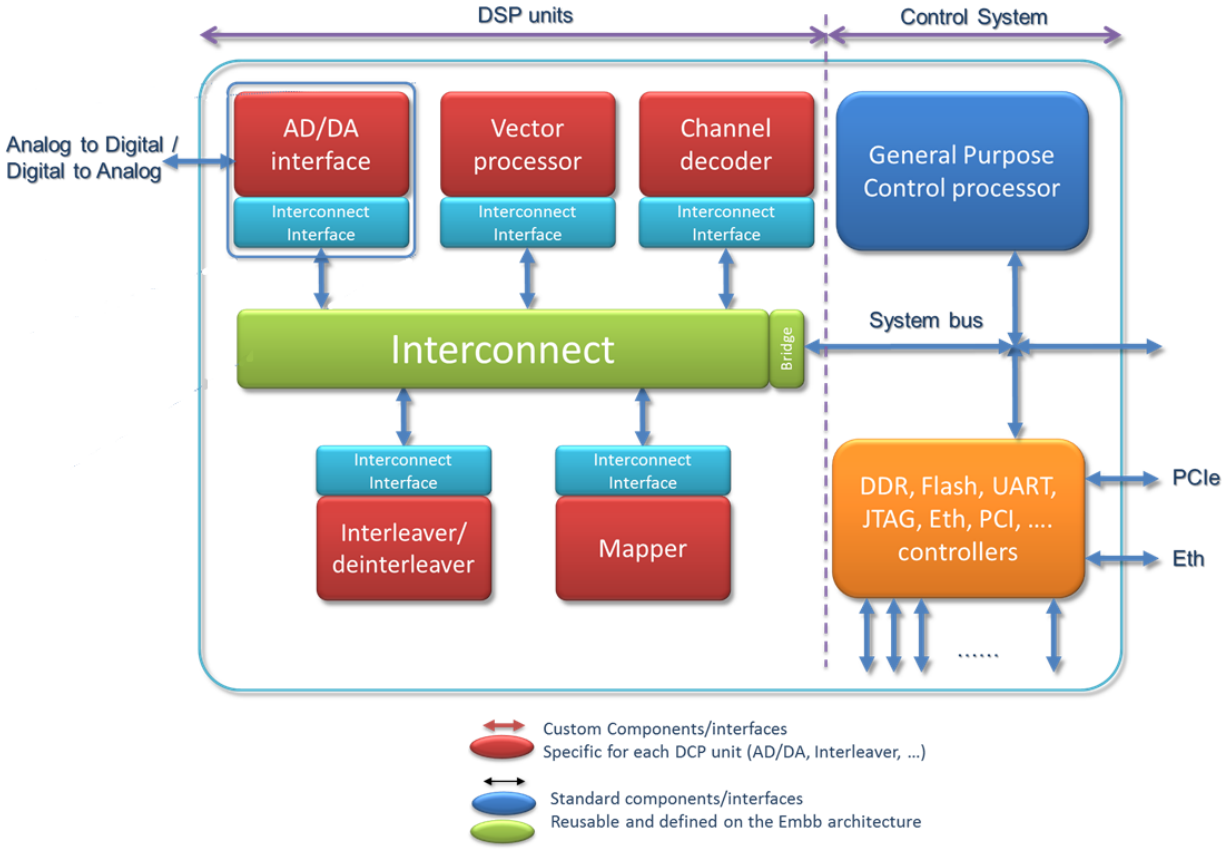
Control



Data processing



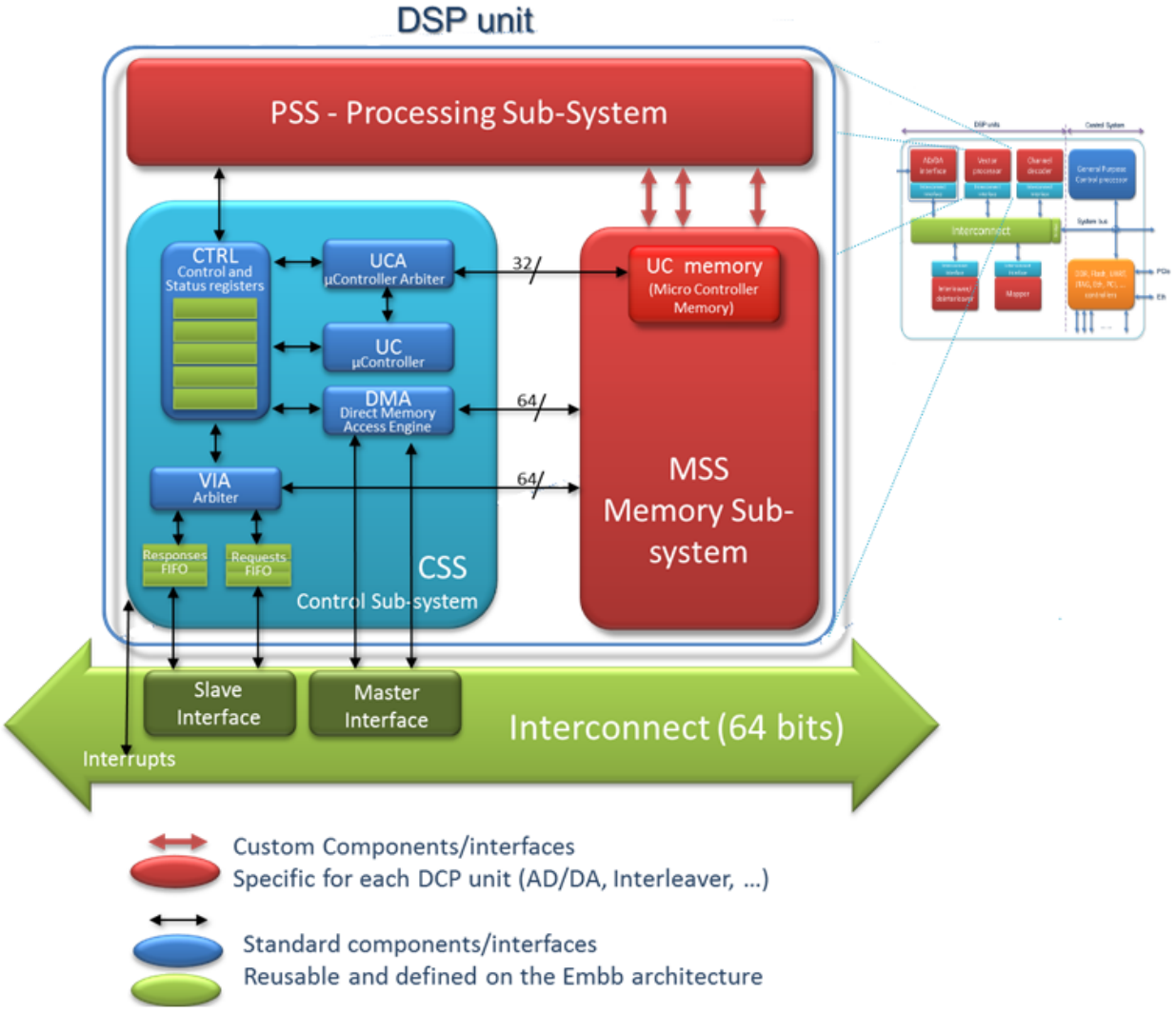
Example: Embb Architecture



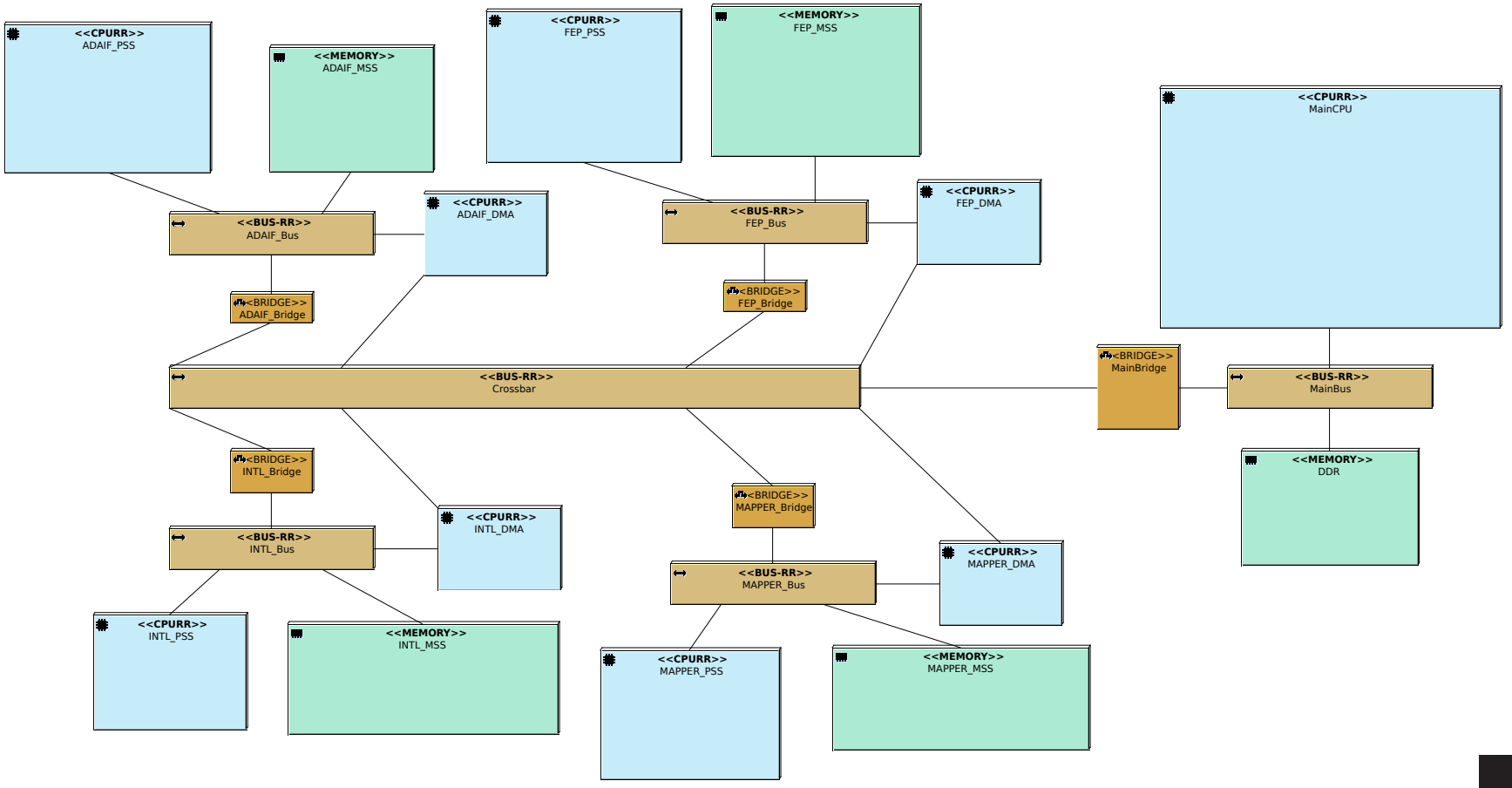
Embb architecture [embb]

[embb] https://gitlab.telecom-paristech.fr/renaud.pacalet/embb

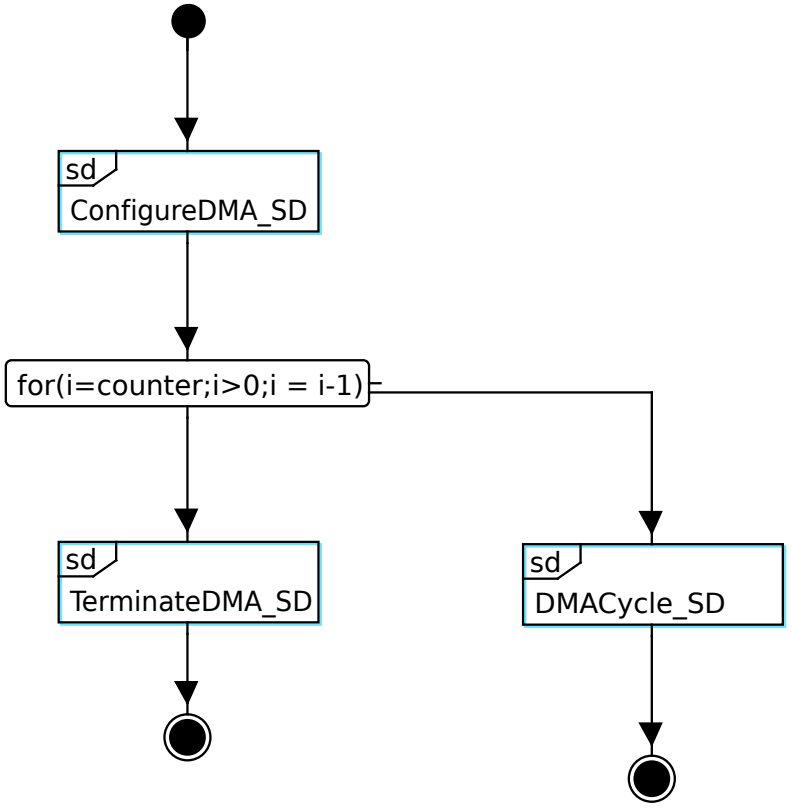
Example: Embb DSP



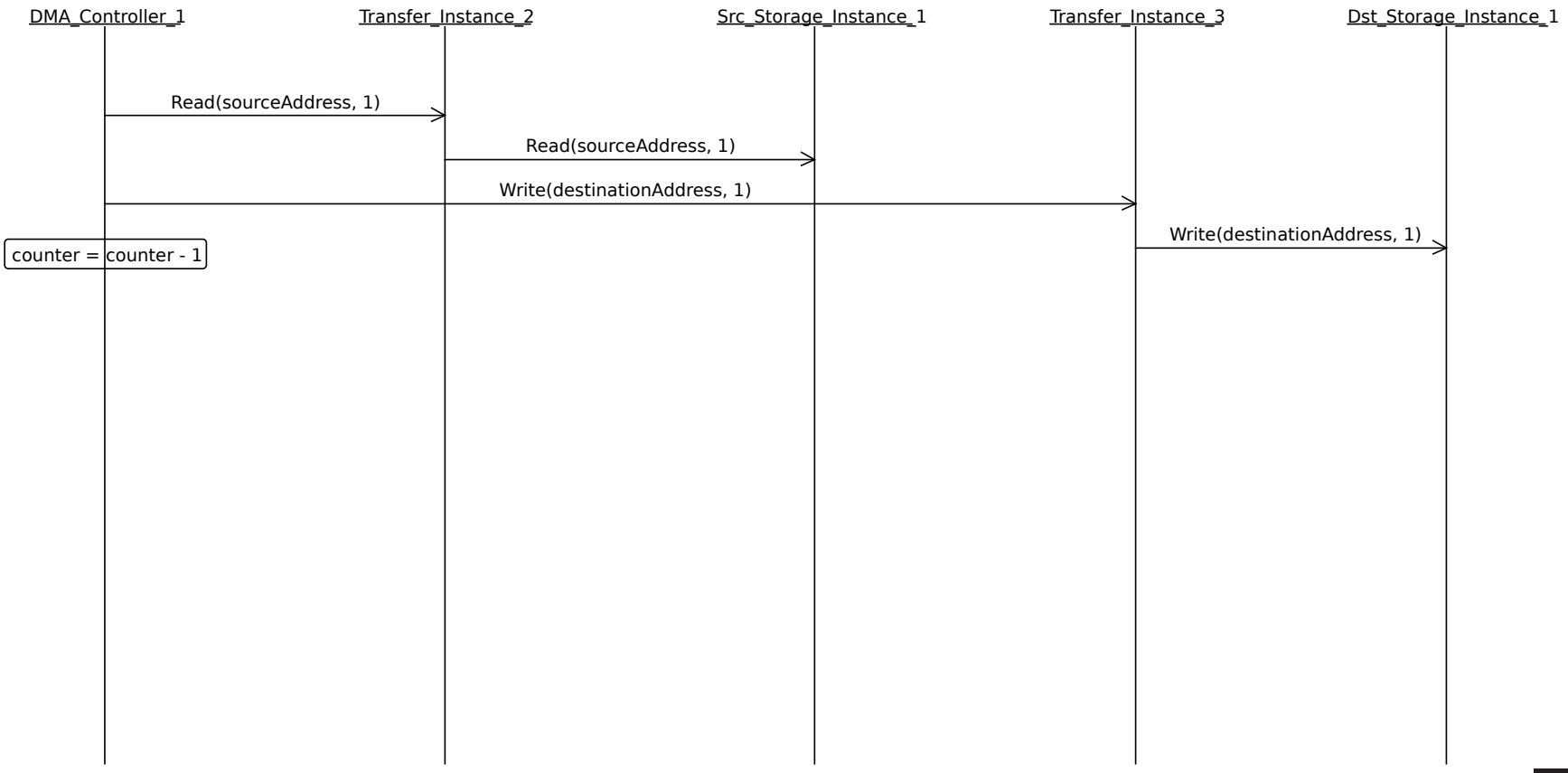
Example: Architecture Model



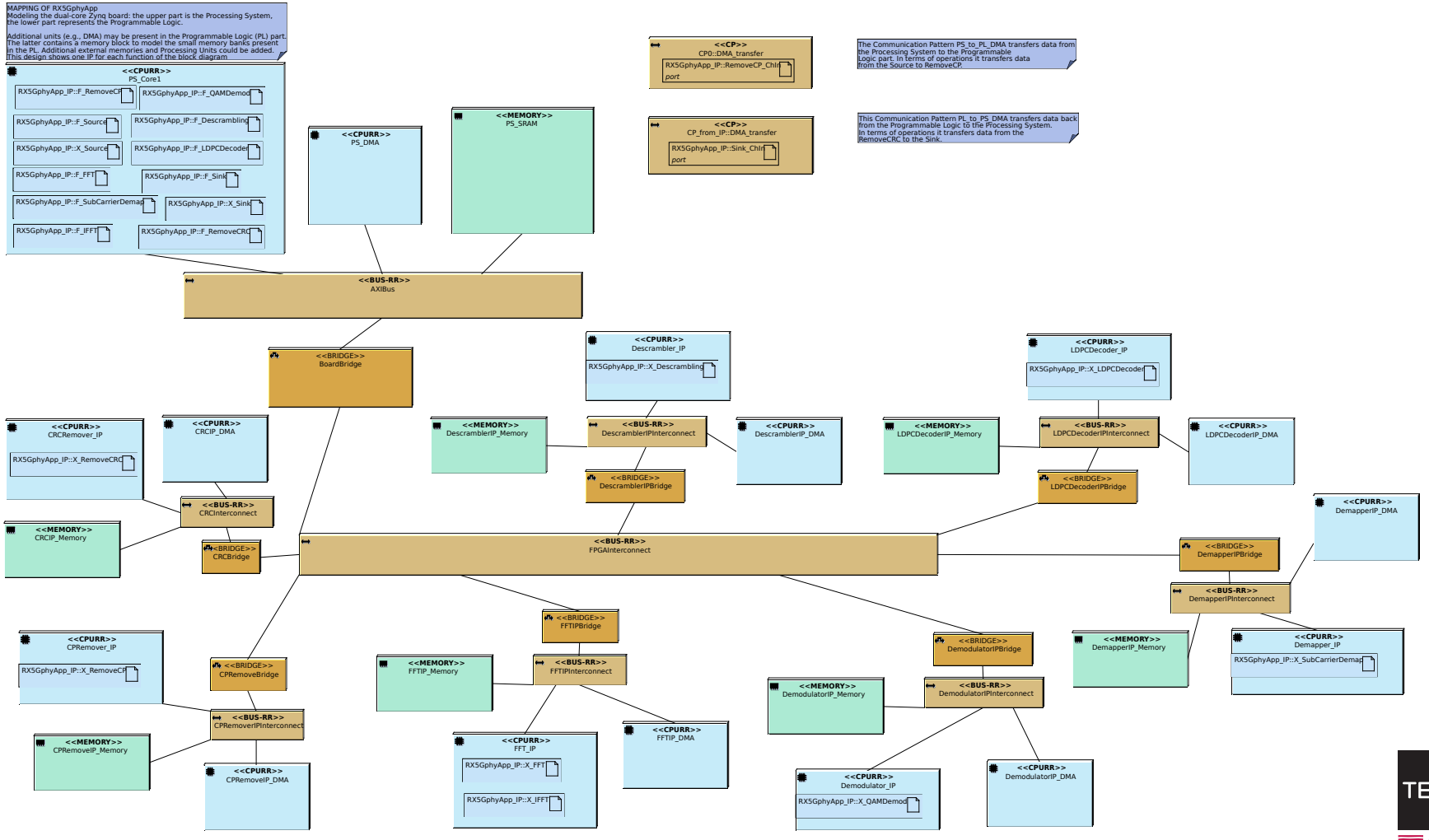
Example: Communication Protocols



Example: Communication Protocols

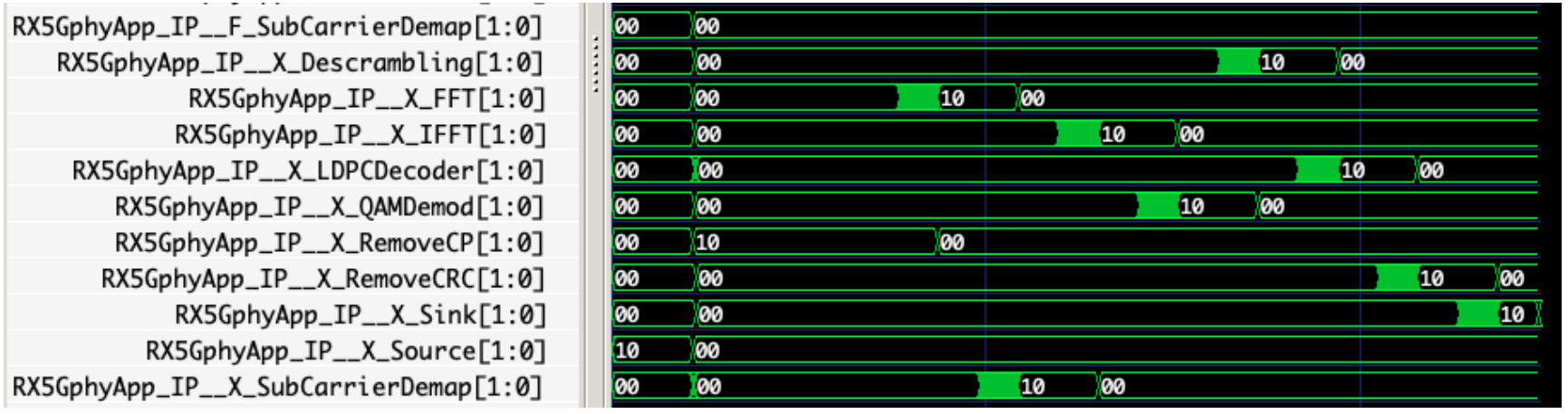


Example: Mapping tasks



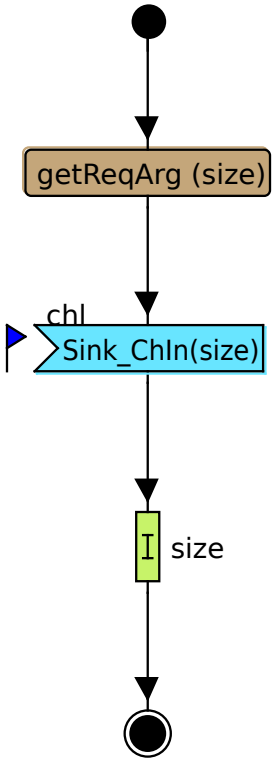
Example: Performance Evaluation

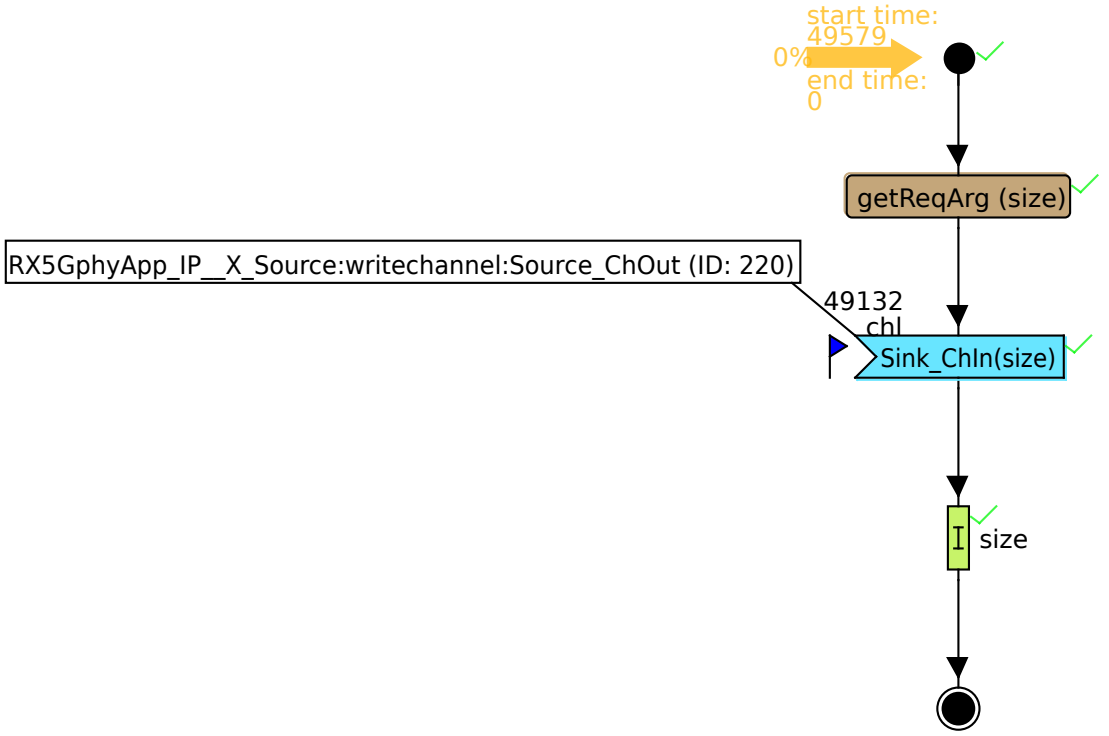
- Interactive simulation (step-by-step, breakpoints, etc.)
- Generation of simulation traces



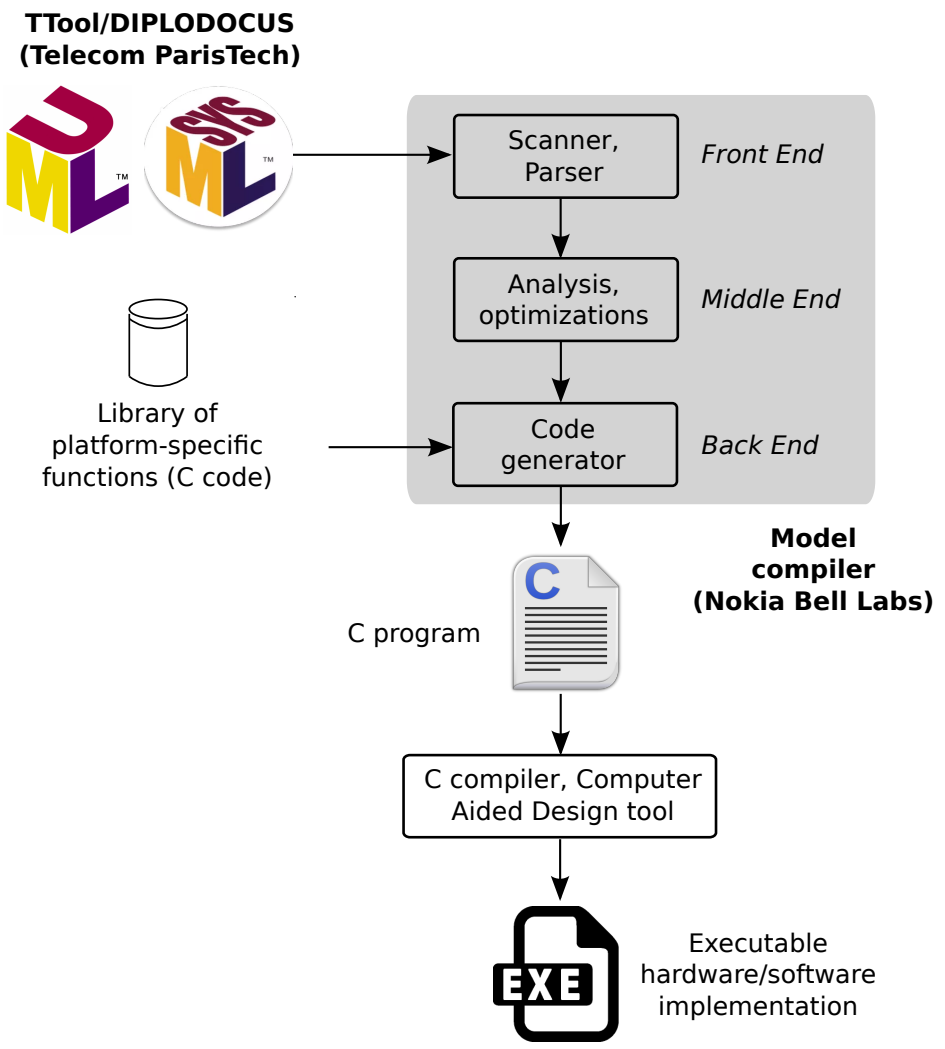
Example: Performance Evaluation

- Latency evaluation





Code Generation



- Full C code
- Front End: Model transformations (SDF, Memory Exclusion Graphs)
- Middle End: Mapping and scheduling of tasks and buffers (Middle-end)
- Back End: C-code generation

Outline

Summary

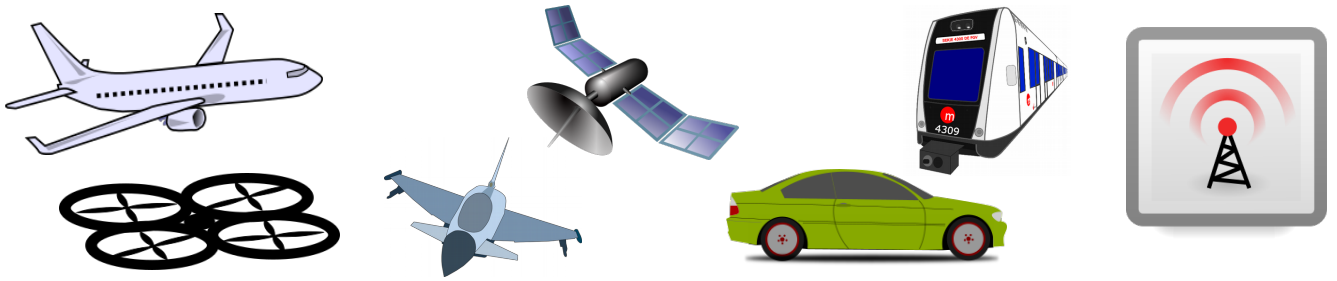
Main features

- Design of dataflow applications
- Push-button approach for verification and code generation
- Safety, security and performance

Domains

- Telecommunications
- Transports
- More generally: embedded systems

TTool: Typical Use



Telecommunication

- Design of 4G and 5G HW/SW architectures (Freescale, Nokia)

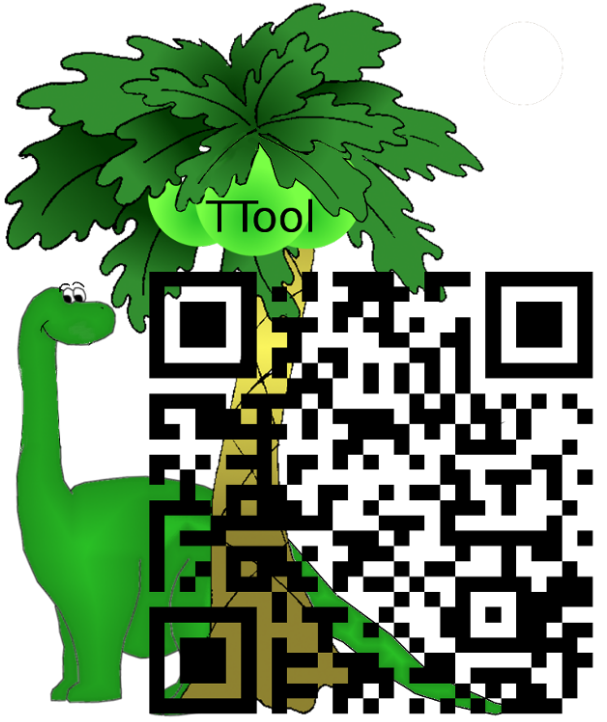
Critical systems

- Motor controllers (Siemens, Continental)

Transports

- Design of safe and secure automotive systems (VEDECOM)
- Security and performance of automotive systems (EVITA - BMW)
- Drones / robots
- Railway systems

Thank You!



References

TTool: ttool.telecom-paristech.fr

Personal website:
http://perso.telecom-paristech.fr/~apvrille

A. Enrici, L. Apvrille, R. Pacalet, "A Model-Driven Engineering Methodology to Design Parallel and Distributed Embedded Systems", ACM Transactions on Design Automation of Electronic Systems (TODAES), Volume 22 Issue 2, January 2017