Tomasz Włostowski

White Rabbit Node Core

Distributed hard realtime with White Rabbit

BE-CO Technical Committee
Genève, 12 March 2015

Outline

- Introduction & Background
- Design of WR Node Core
- Project status
- Future outlook

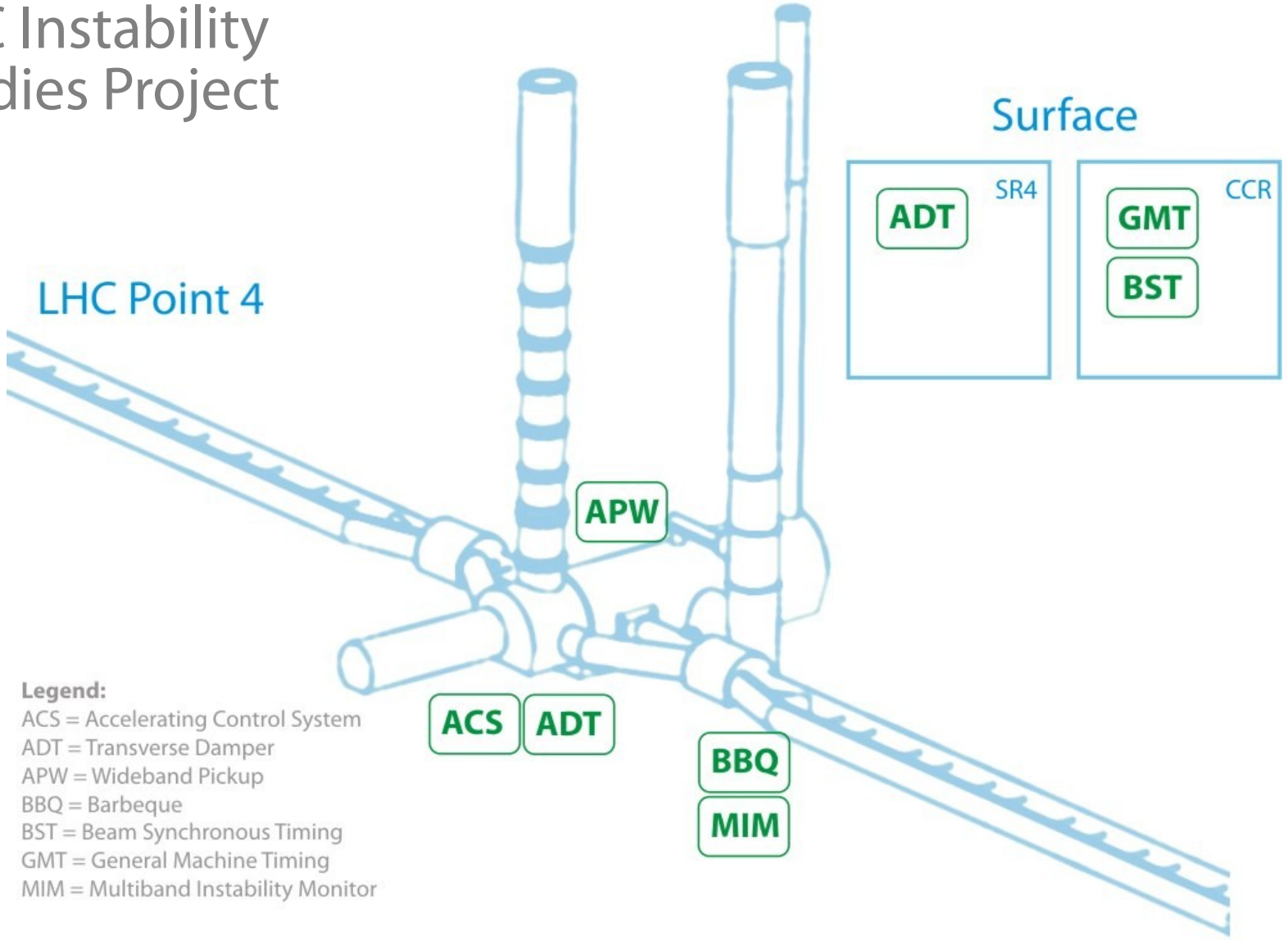
Background

LHC Instability Studies Project

- Instruments in point 4 detect an onset of beam instability
- Generate a trigger
- Distribute the trigger to other instruments and acquire a massive amount of data for offline study

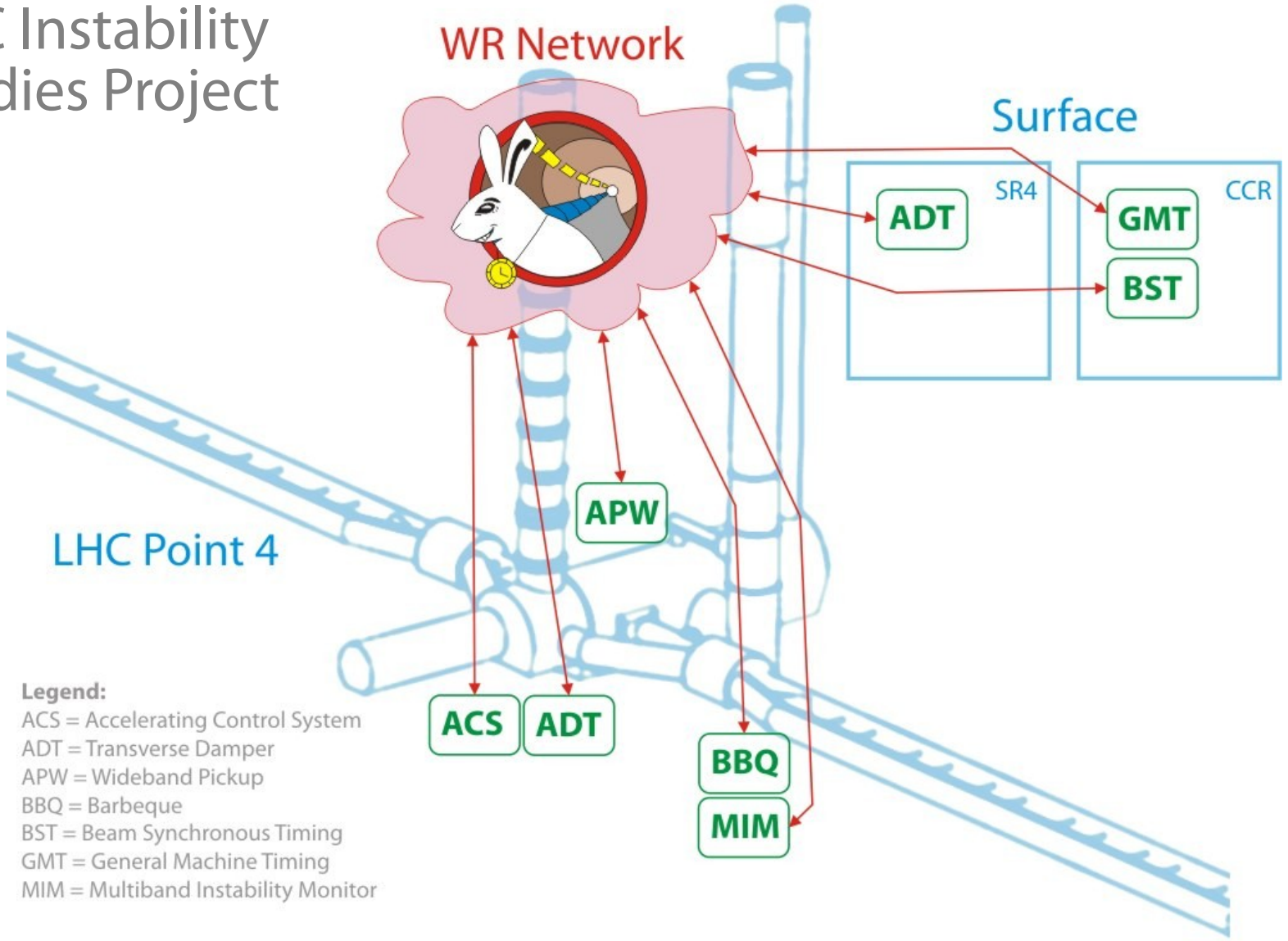
Background

LHC Instability Studies Project



Background

LHC Instability Studies Project



The Challenge

- Exchange the triggers between any pair of devices in the network
- Never exceed design latency (max 270 us)
- Never miss a trigger

Distributed hard realtime system

Why WRNC?

- Hard realtime not possible on PCs
- PCs are not deterministic enough
- Impossible to meet LIST latency requirements

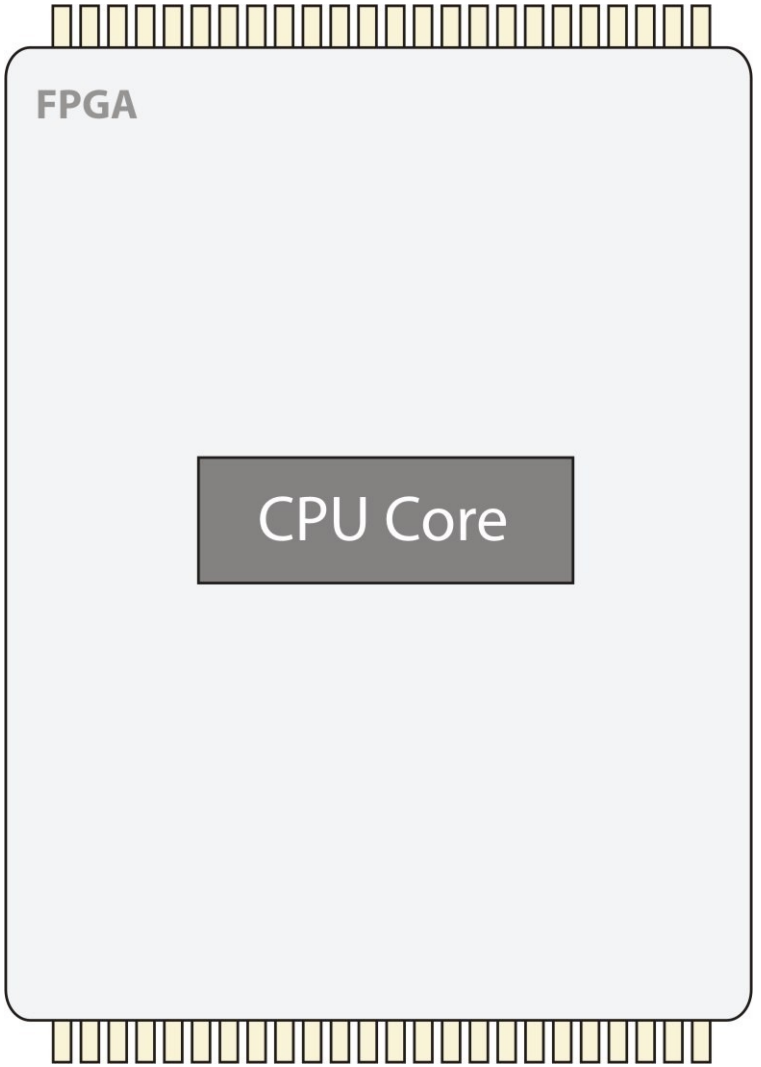
Make a computing system deterministic by design!

The idea of WRNC

- **Take an FPGA chip**
- Take a CPU core that is deterministic
- Put as many CPUs as needed
- Let them communicate with each other...
- … and with the external world
- Connect user cores and hardware

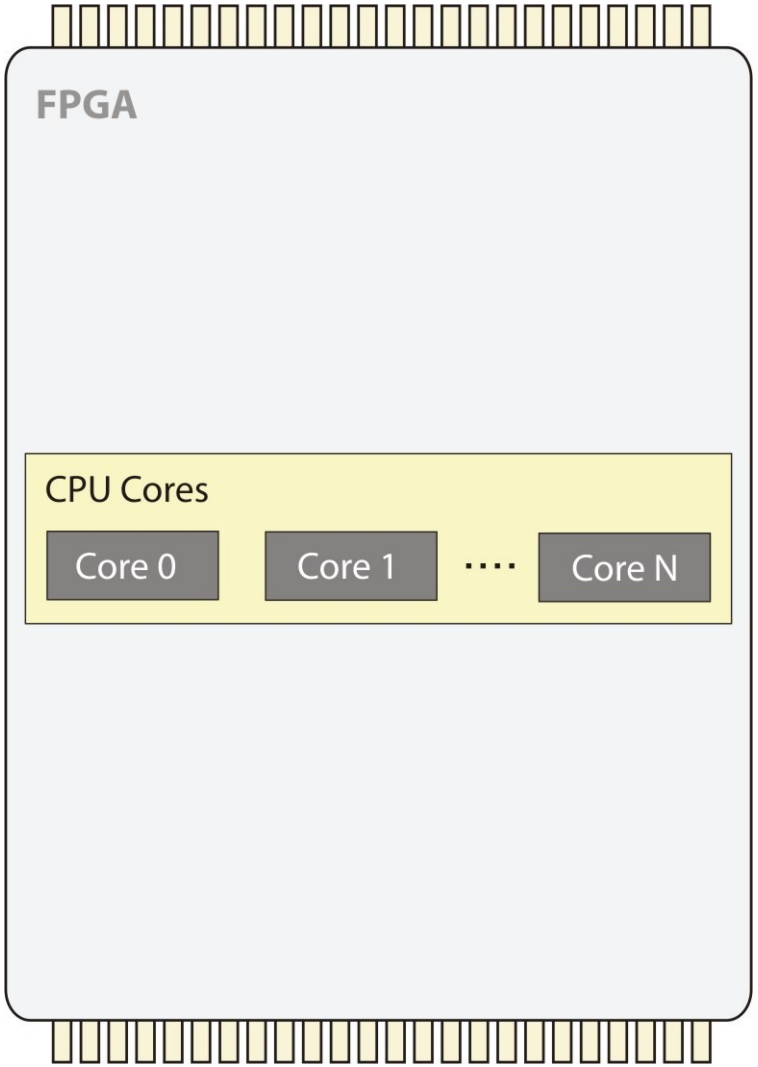
The idea of WRNC

- Take an FPGA chip
- **Take a CPU core that is deterministic**
- Put as many CPUs as needed
- Let them communicate with each other...
- … and with the external world
- Connect user cores and hardware



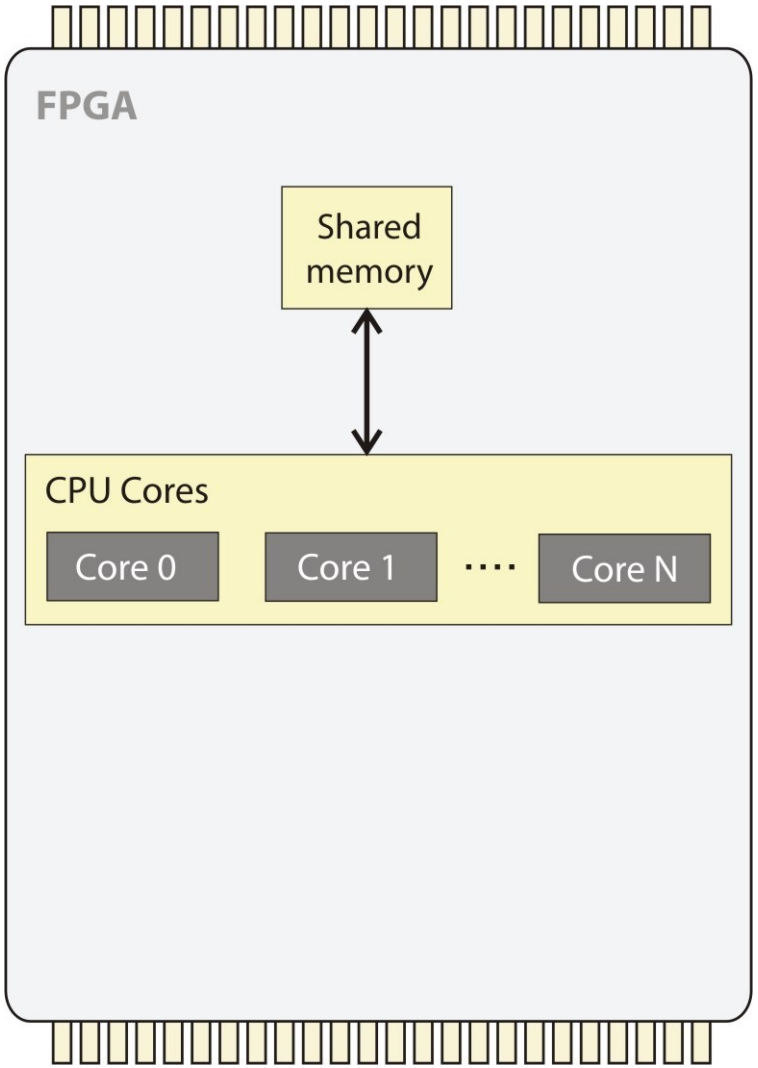
The idea of WRNC

- Take an FPGA chip
- Take a CPU core that is deterministic
- **Put as many CPUs as needed**
- Let them communicate with each other...
- … and with the external world
- Connect user cores and hardware



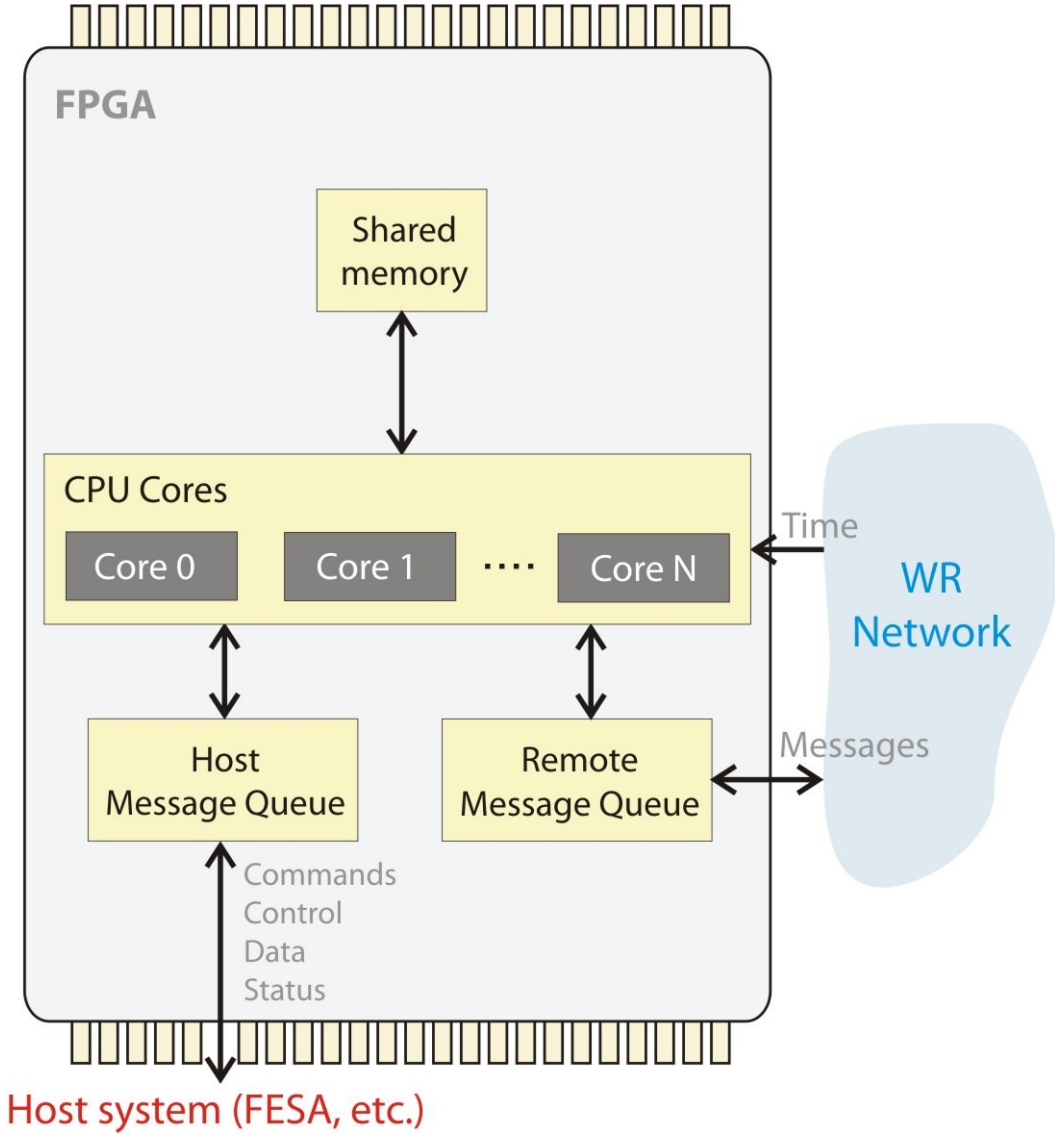
The idea of WRNC

- Take an FPGA chip
- Take a CPU core that is deterministic
- Put as many CPUs as needed
- **Let them communicate with each other…**
- … and with the external world
- Connect user cores and hardware



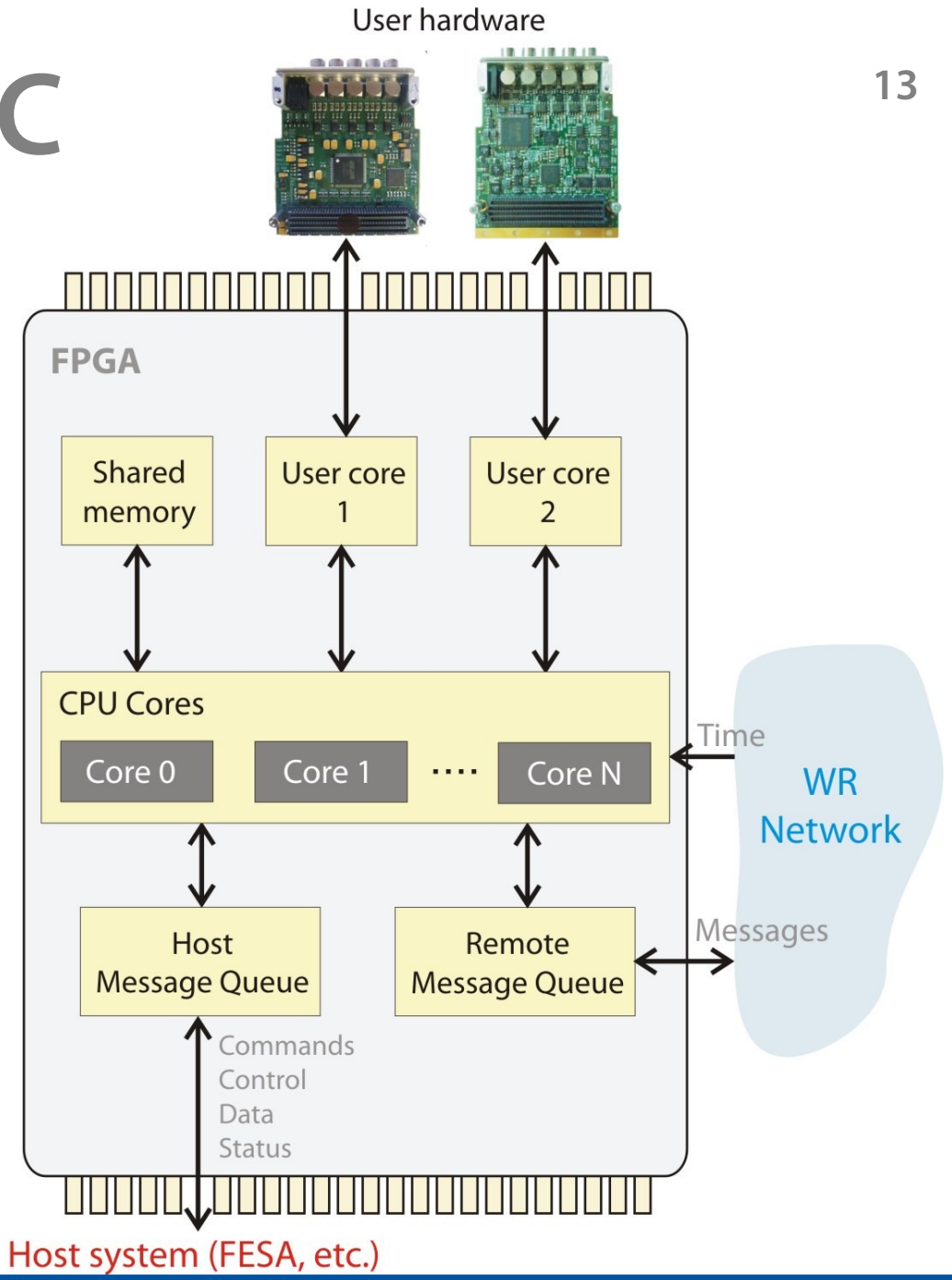
The idea of WRNC

- Take an FPGA chip
- Take a CPU core that is deterministic
- Put as many CPUs as needed
- Let them communicate with each other...
- **… and with the external world**
- Connect user cores and hardware



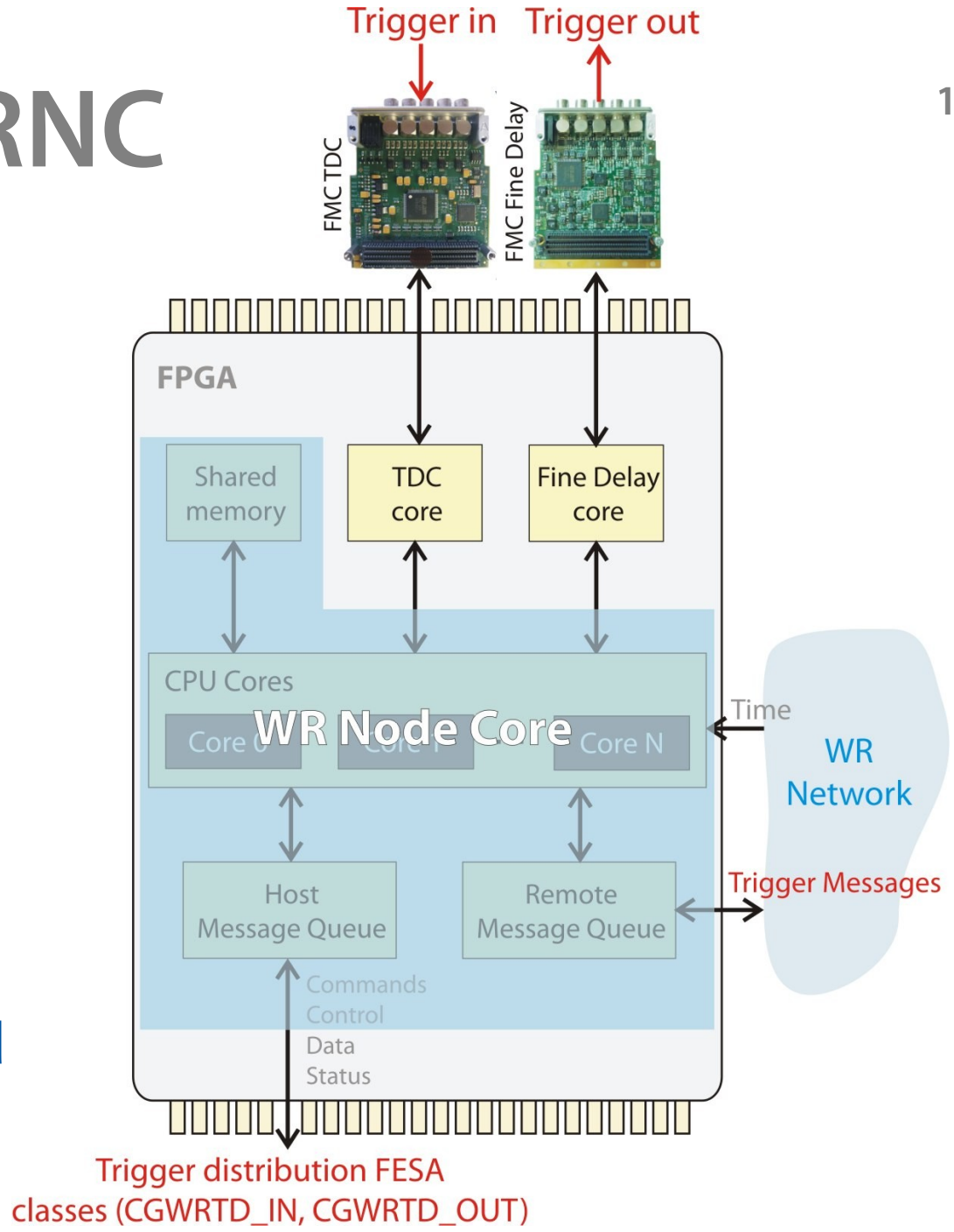
The idea of WRNC

- Take an FPGA chip
- Take a CPU core that is deterministic
- Put as many CPUs as needed
- Let them communicate with each other...
- … and with the external world
- **Connect user cores and hardware**



The idea of WRNC

- Take an FPGA chip
- Take a CPU core that is deterministic
- Put as many CPUs as needed
- Let them communicate with each other...
- … and with the external world
- **Connect user cores and hardware**



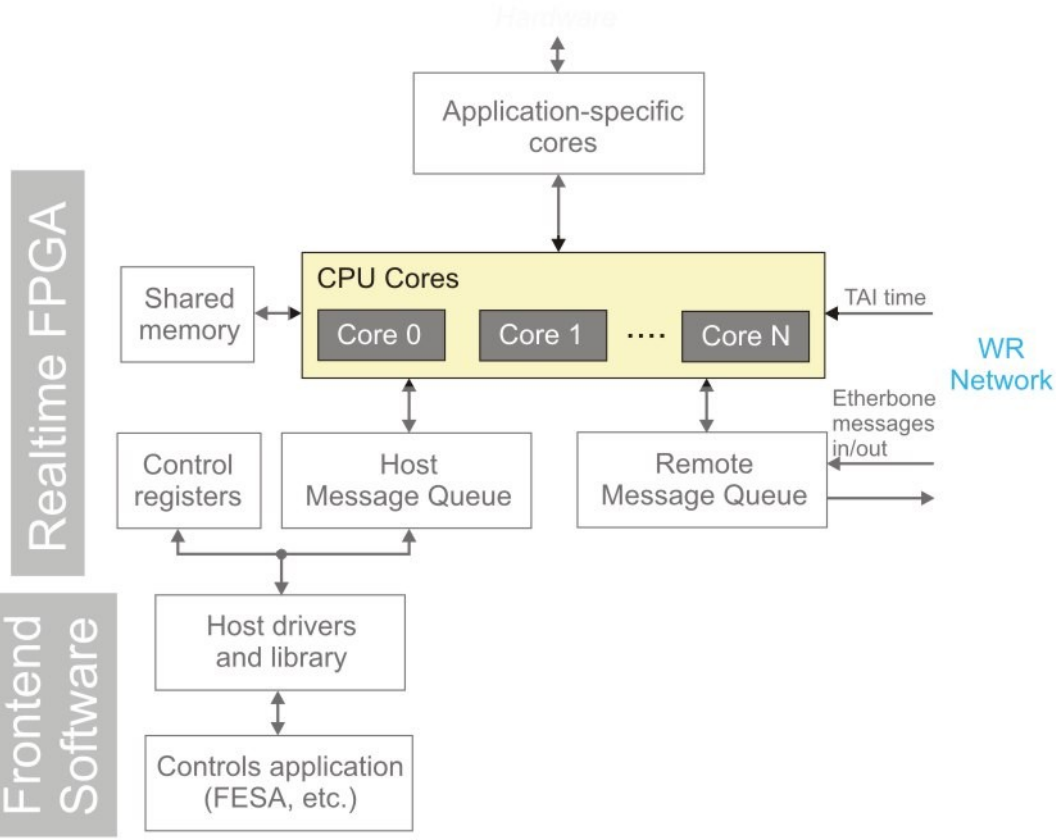
Our internal (HT) reasons

- Variety of hard real time systems under HT responsibility
- Determinism (in most cases) is more important for us than speed
- Programming in C takes less time and effort than HDL design
- One generic core instead of many custom-designed devices

Build your hard RT system from simple design blocks

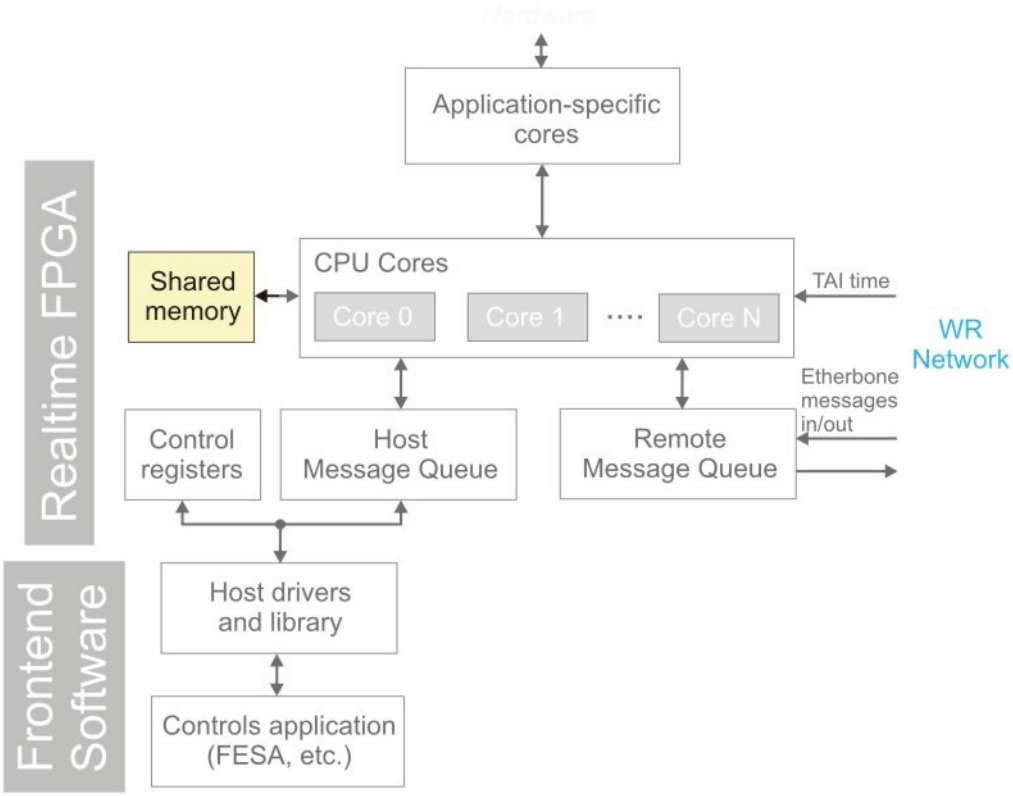
Technicalities
The CPUs

- Up to 8 LM32 cores, with private code/data memory
- Programed in *bare metal* C, using standard GCC tool chain (including debugger)
- No caches, no interrupts (to ensure determinism)
- Inter-core communication through shared memory
- Program loading and flow control from the host system
- Run with same time base thanks to WR



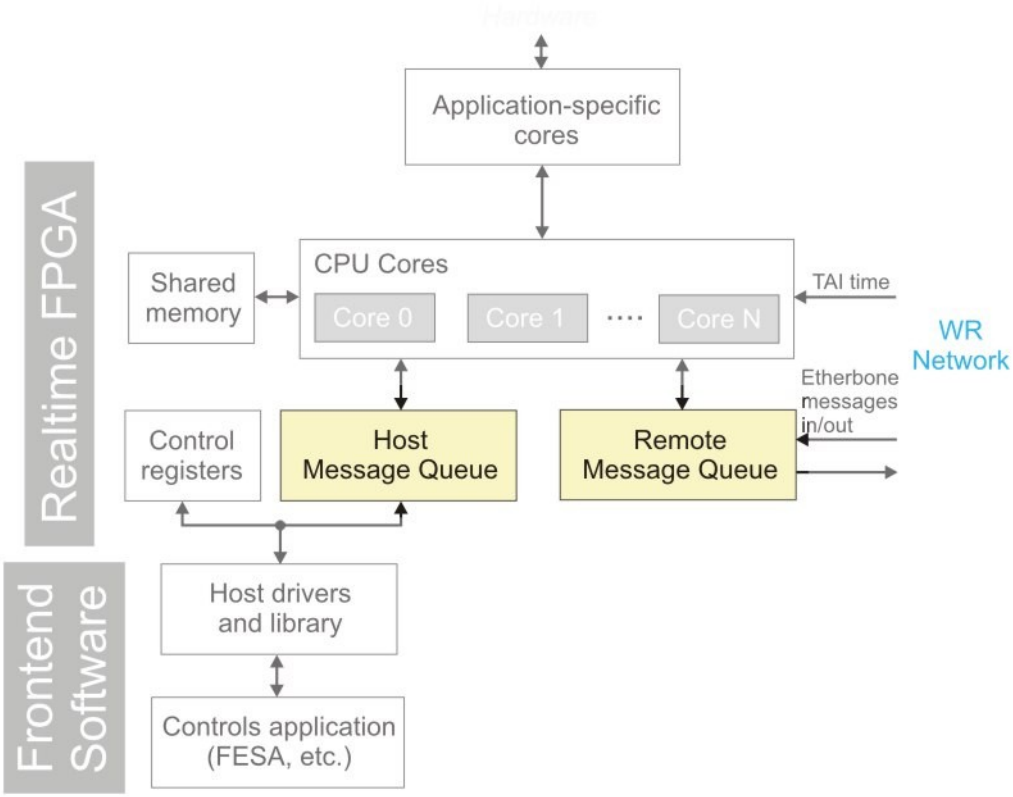
Technicalities
Shared memory

- Foreseen for multiprocessor communication and task synchronization
- Small, but atomically accessed
- Task synchronization primitives – mutexes, semaphores, queues, flags, events



Technicalities
Message queues

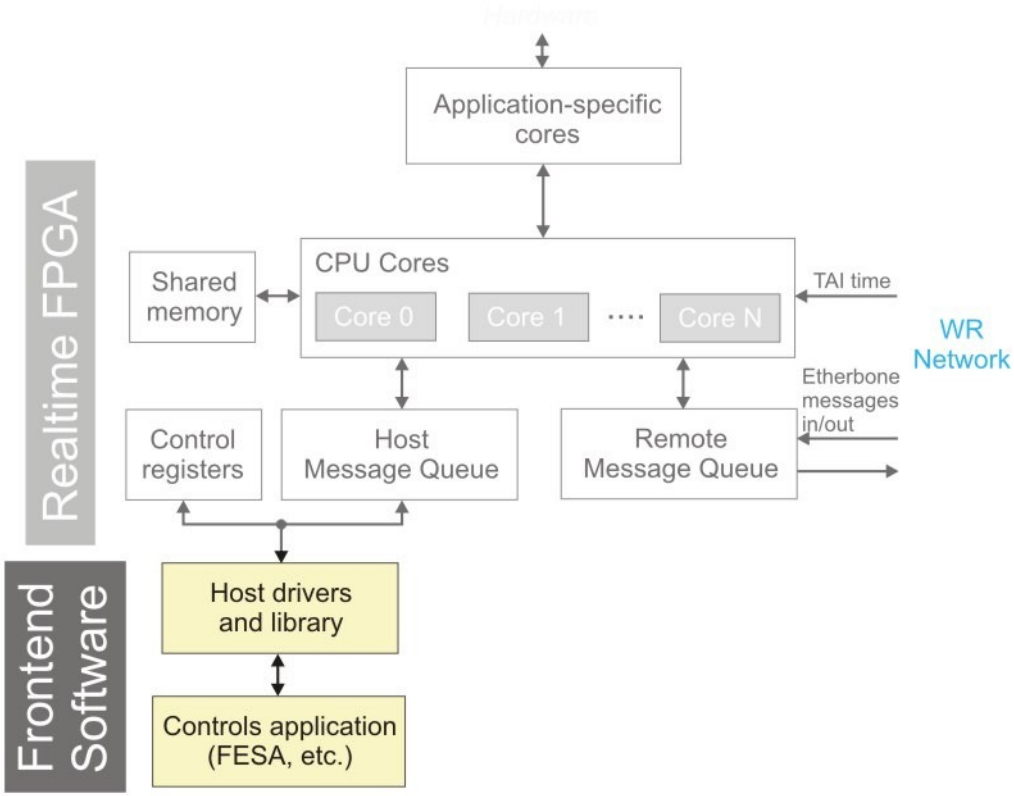
- Simple FIFO queues holding multiple messages
- Host Message Queue for communication with the frontend software
- Remote Message Queue for communication with other WR Nodes over WR network



Technicalities

Frontend software & drivers

- Single generic kernel driver + user space C library
- Applications can be written in user space only → easier development
- Host Message Queue and Shared Memory access
- Loading CPU applications and controlling each core's execution flow
- Python and C++ APIs (future)



Status

- First version of WRNC ready (gateware & drivers)
- Latency and determinism goals achieved
- LIST being deployed in the LHC, fully operational in 6 months

Future outlook
Pending WRNC-based projects

- Trigger distribution in OASIS (before LS2)
- Distributed DDS system for RF transmission over Ethernet (Q2 2015)
- New *WorldFIP* master design based on WRNC (Q4 2015)
- Proposal for future timing for CERN: GMT and BST in the same network (post LS2)

Conclusion

Proven hard realtime performance

Simple to use and program

WRNC as a general service for developing hard realtime applications at CERN

(with or without White Rabbit)

Questions?

Demo time!

Let's write a simple program in C that:

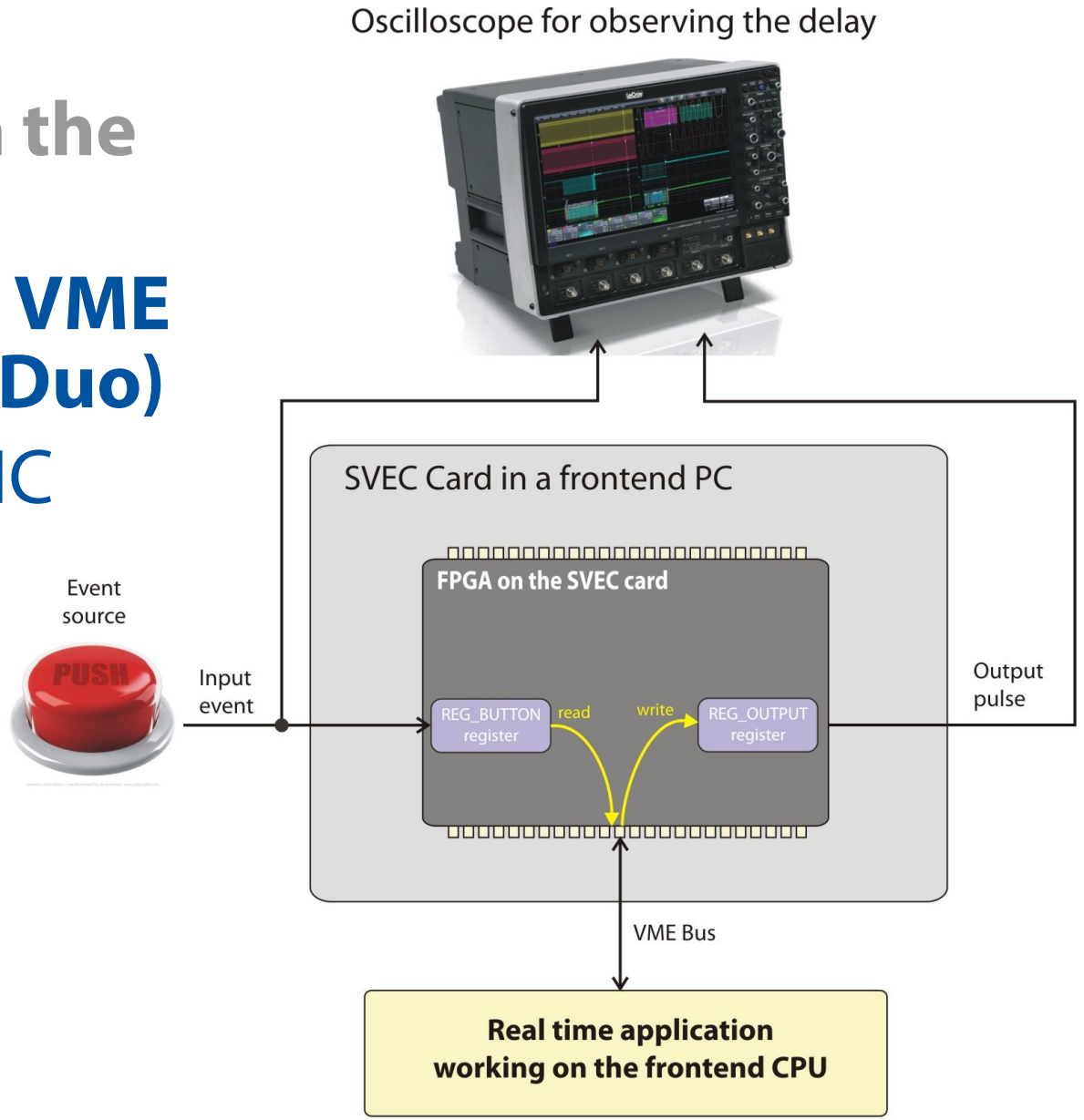
- Waits for a button to be pressed
- Emulate some calculations (fixed delay loop)
- Produces a pulse on an output of a VME card

And observe its reaction time to input events with an oscilloscope!

Demo time!

Compile and run the program on:

- **CPU of the CERN VME frontend (Core2 Duo)**
- CPU inside a WRNC



Demo time!

Compile and run the program on:

- CPU of the CERN VME frontend (Core2 Duo)
- **CPU inside a WRNC**