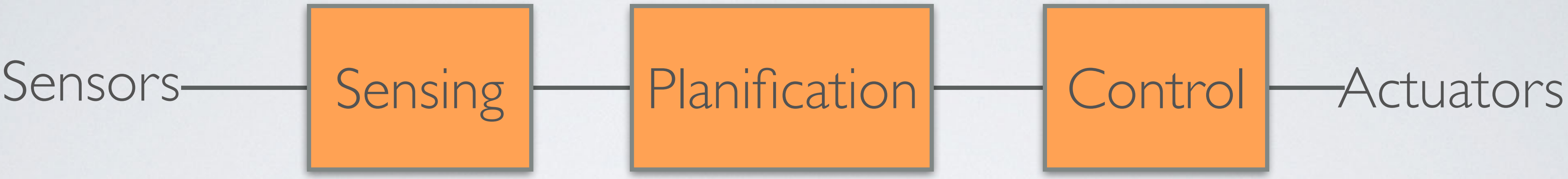
TOWARD A METHODOLOGY TO TURN SMALLTALK CODE INTO FPGA

LE Xuan Sang[1,2]
Loïc LAGADEC[1], Luc FABRESSE[2],
Jannik LAVAL[2] and Noury BOURAQADI[2]

[1] Lab-STICC, ENSTA Bretagne
[2] Institut Mines-Telecom, Mines Douai

Robotic application requirements



- Robotics applications demand :
  - Real-time + amount of data : Processing power
    Example in vision : 20 images of 320x240/s ≃ 37Mbps.
  - Flexibility to evolution and unforeseen change of hardware
    (adding more device/sensor/actuators, improvement of circuit etc.)

Smalltalk in Robotic Software Development

- Simplicity & rich semantic
- High-level abstraction
- Smalltalk is also an IDE:
  - Support Agile methodology
  - Valuable ability of debugging & testing application
- **BUT :** Time-consuming for mass data processing

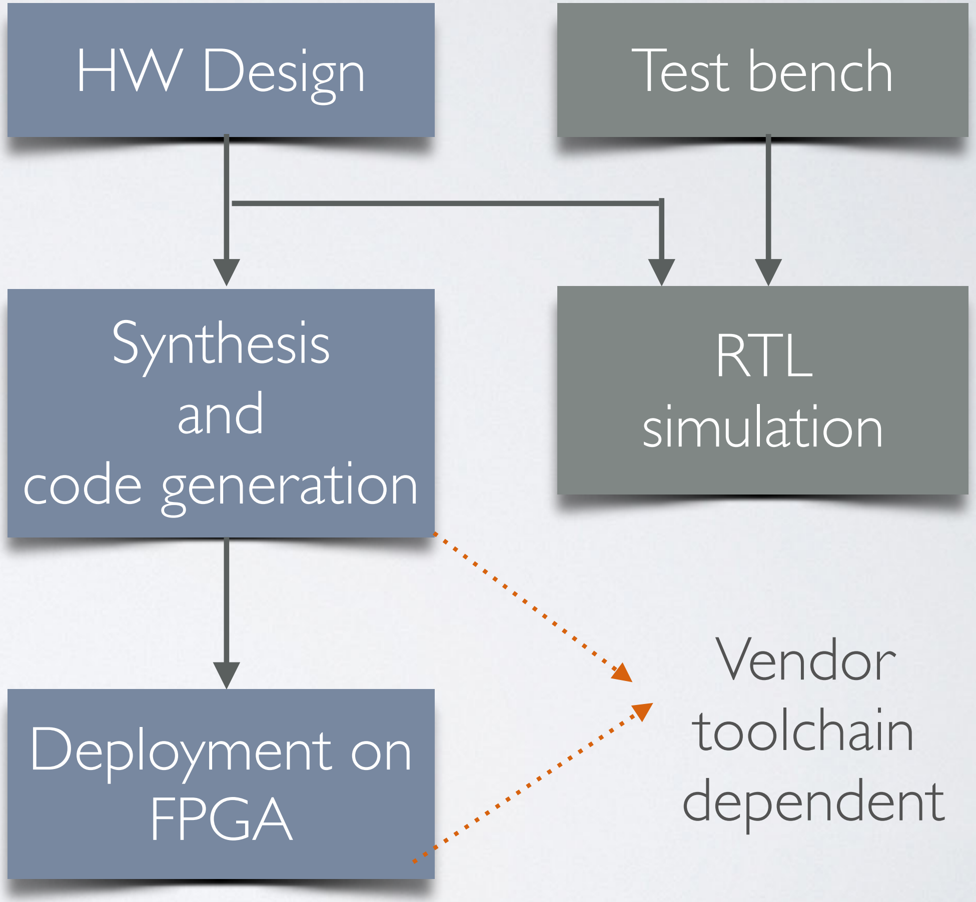
Field Programmable Gate Array (FPGA)

- Configurable hardware/chip

- Parallel processing
- FPGA circuits are designed using Hardware Description Language (HDLs).
- **BUT:** HDL-based design is unsuitable for hardware/software co-design

FPGA HDL-BASED DESIGN

1. Hardware Design (HDL)
2. Write tech-bench and perform Register Transfer Level simulation (RTL)
3. Synthesis and code generation
4. Deployment on FPGA



FPGA HDL-BASED DESIGN

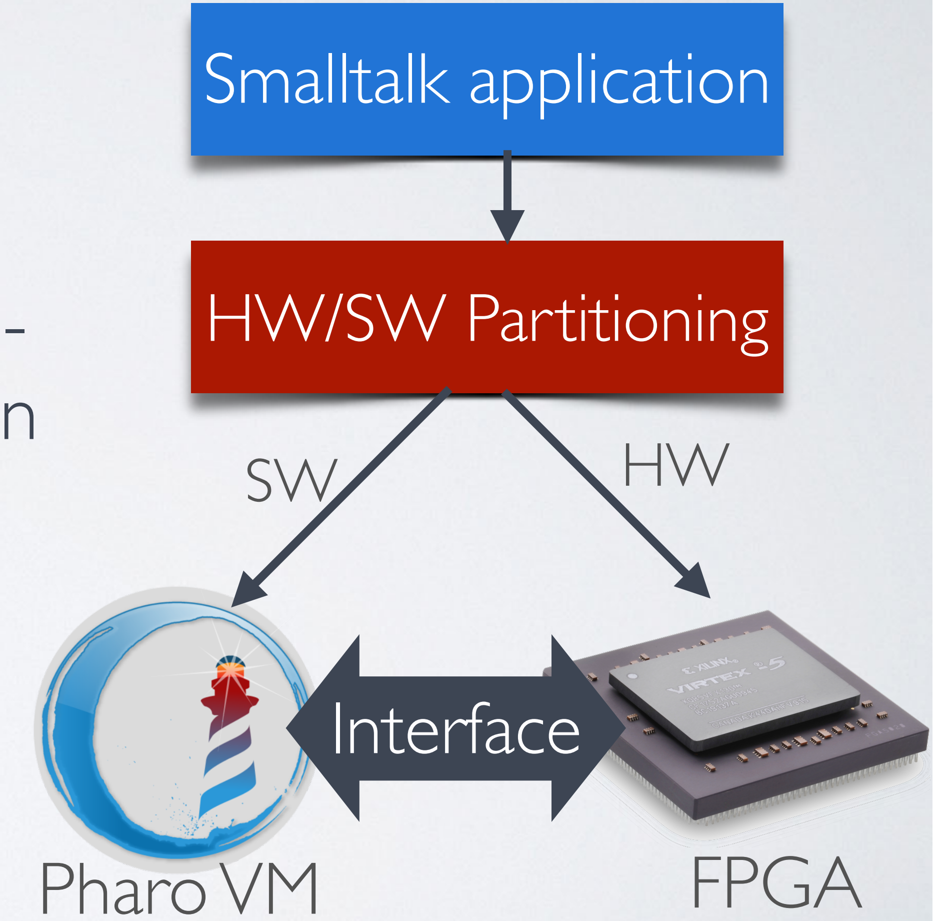
- **HDLs support specification up to Register Transfer Level (RTL)**. **But** :
  - Lack abstractions to implement high-level algorithms
  - Debugging is really hard (waveforms)
  - Not adequate for high-level modelling/programming
  - Hardware dependency → limit reusability

OBJECTIVE

Using Smalltalk in hardware/ software co-design

- **Hardware :** Smalltalk as a high-level description and verification language

- **Software** : Smalltalk robotic application that interact with FPGA

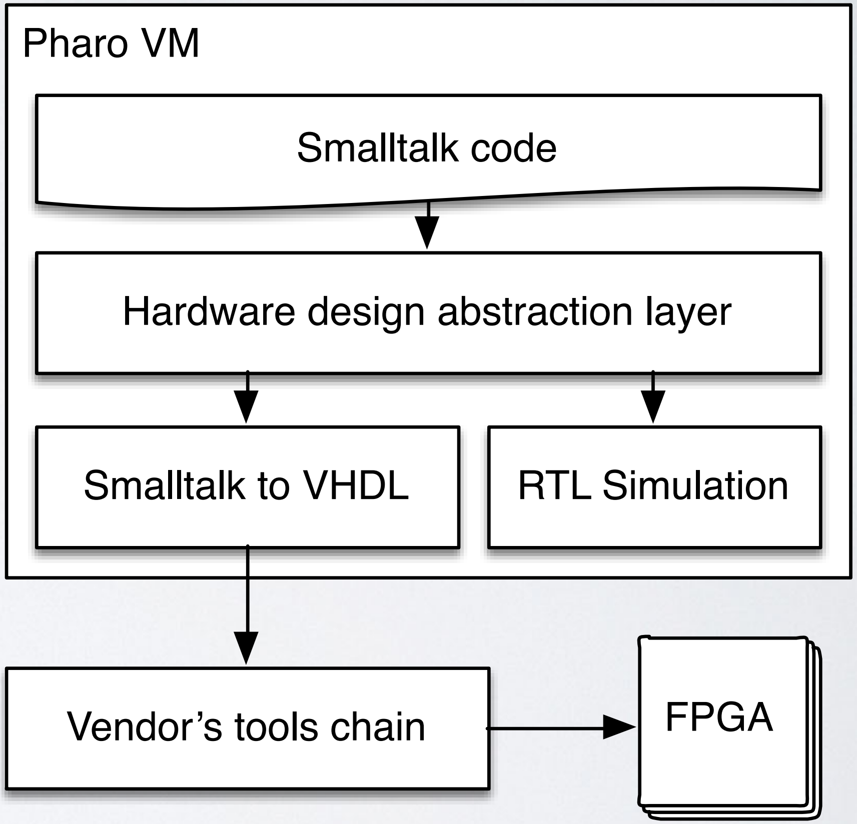


Smalltalk application

HW/SW Partitioning

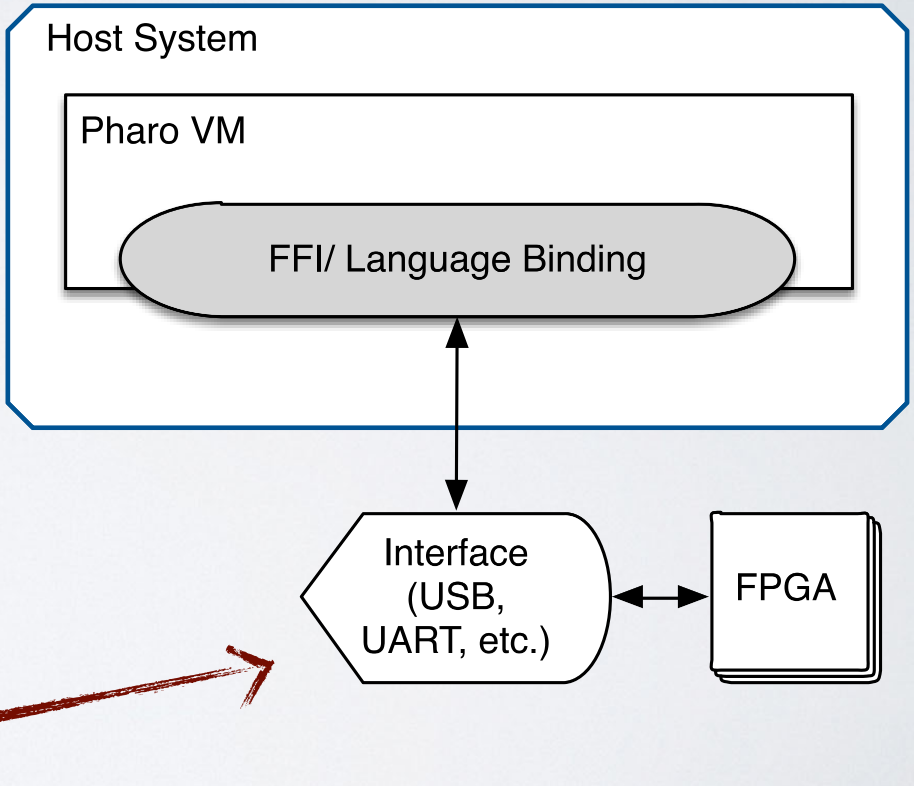
SMALLTALK FOR HARDWARE DESIGN

- **Smalltalk-based design**
  - Hardware design abstraction layer
  - RTL simulation
    - Waveform tracing
    - Unit Test
  - Smalltalk-VHDL conversion
  - Reusability & extensibility
  - Vendor's tool interaction



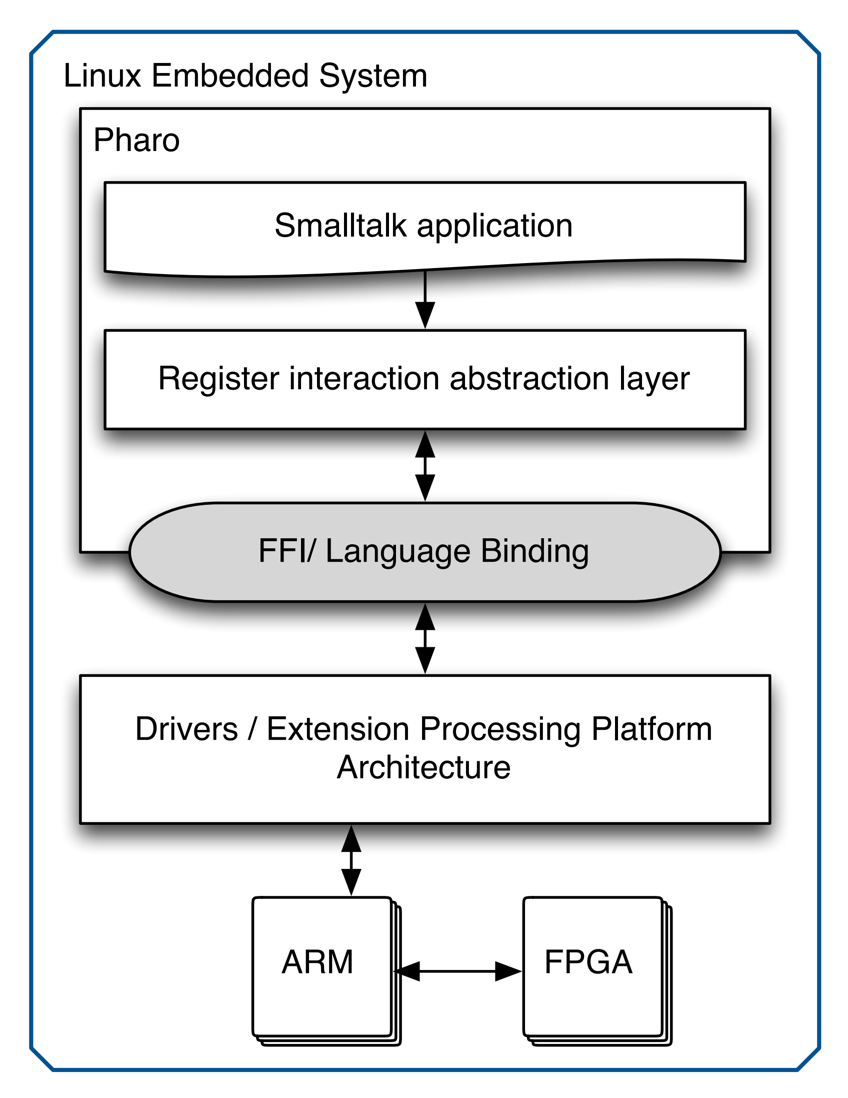
SMALLTALK FOR SOFTWARE PROGRAMMING ON FPGA (1)

- **Standalone FPGA**
  - Operate independently with the host system
  - Communicate with the host system via interfaces of communication : USB, RS232, etc.
  - Smalltalk application talk to FPGA via Foreign Function Interfaces (such as Native Boost)
- **Problem :** bandwidth bottleneck



SMALLTALK FOR SOFTWARE PROGRAMMING ON FPGA (2)

- **FPGA-ARM SoC/SoM**
    - FPGA System on Chip/Module : all in one system
    - Accelerate FPGA - ARM communication →reduced bottleneck.
    - Direct access to FPGA registers via system library
    - Smalltalk application talk to FPGA registers via Register interaction abstraction layer which uses FFI


Linux Embedded System
Pharo
Smalltalk application
Register interaction abstraction layer
FFI/ Language Binding
Drivers / Extension Processing Platform Architecture
ARM
FPGA

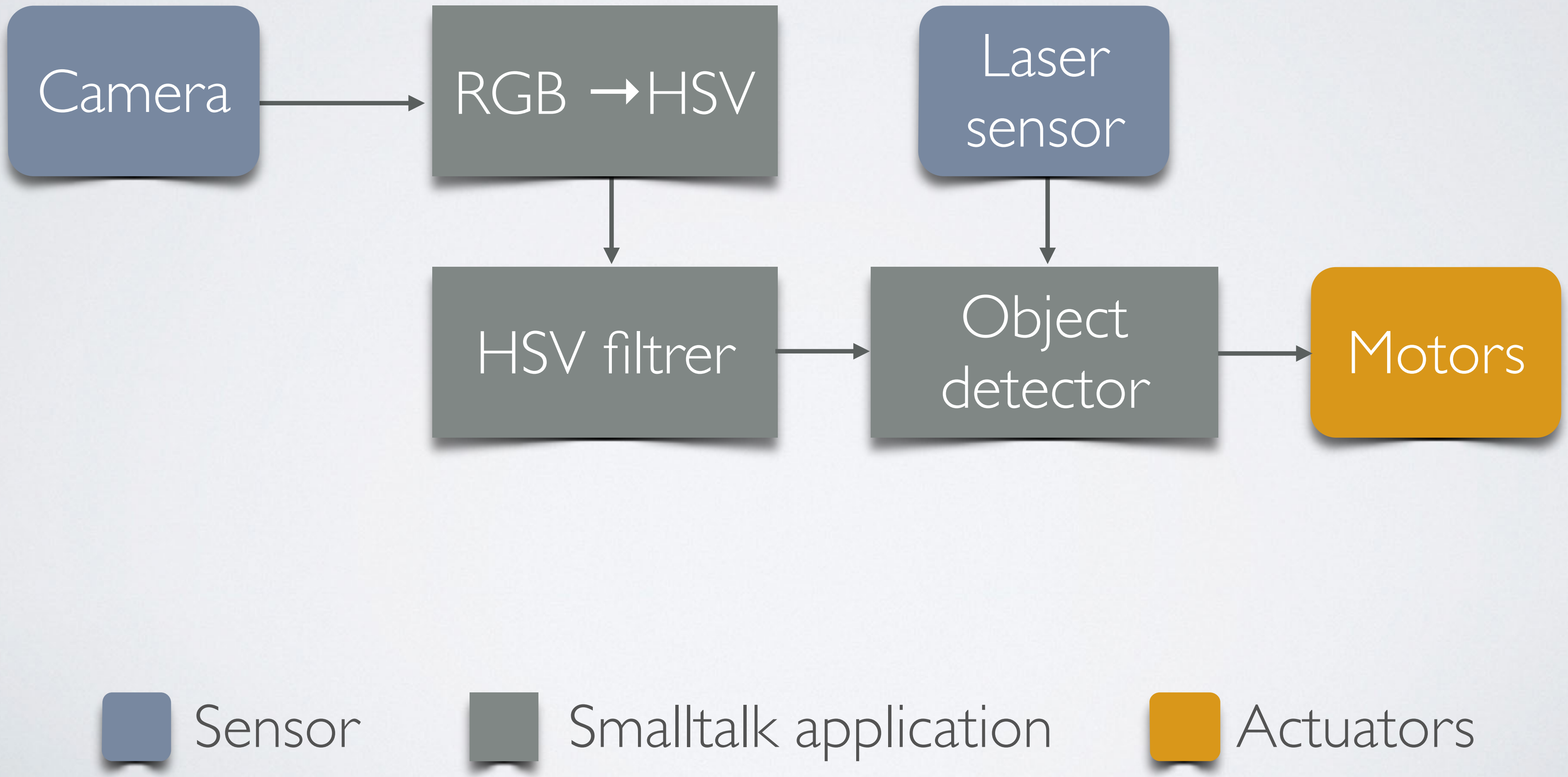
EXPERIMENT

- Build a Pharo robotic application
- Identify critical parts
- Project the critical parts on FPGA
- Evaluation of performance gain/loss

VIDEO

http://car.mines-douai.fr/2014/04/pharos-based-tracker-robot/

EXPERIMENT



Camera

RGB →HSV

Laser sensor

HSV filtrer

Object detector

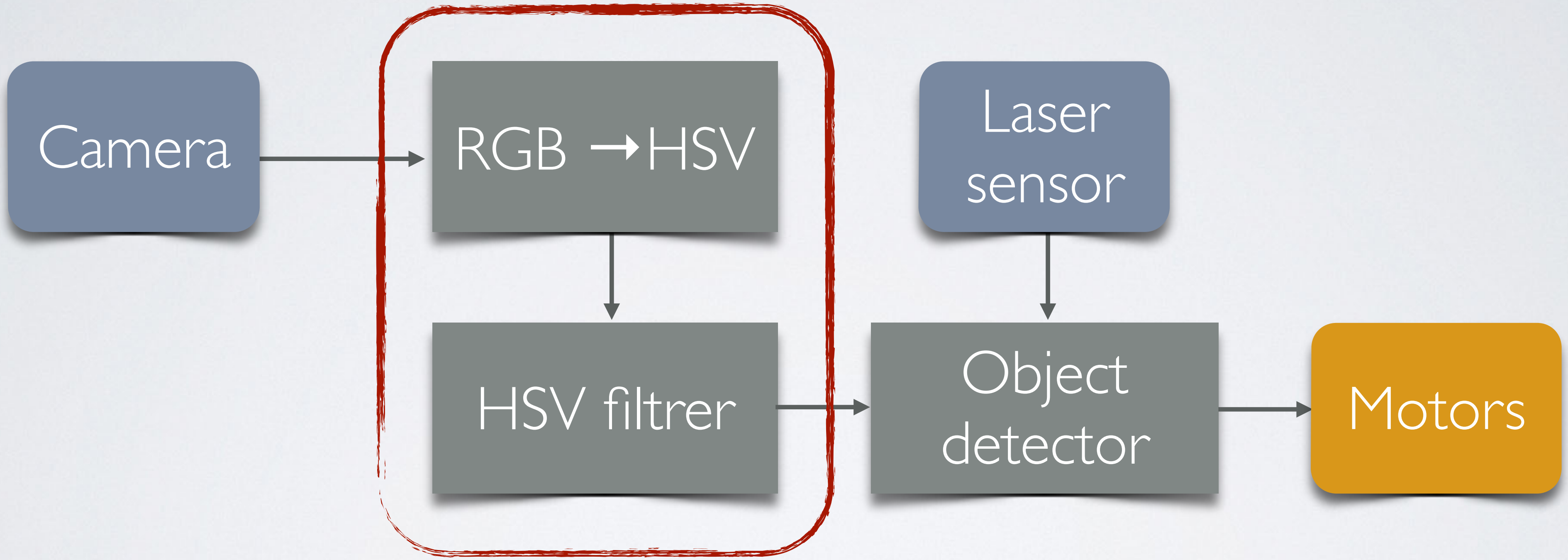
Motors

Sensor

Smalltalk application

Actuators

EXPERIMENT



Camera

RGB →HSV

HSV filtrer

Laser sensor

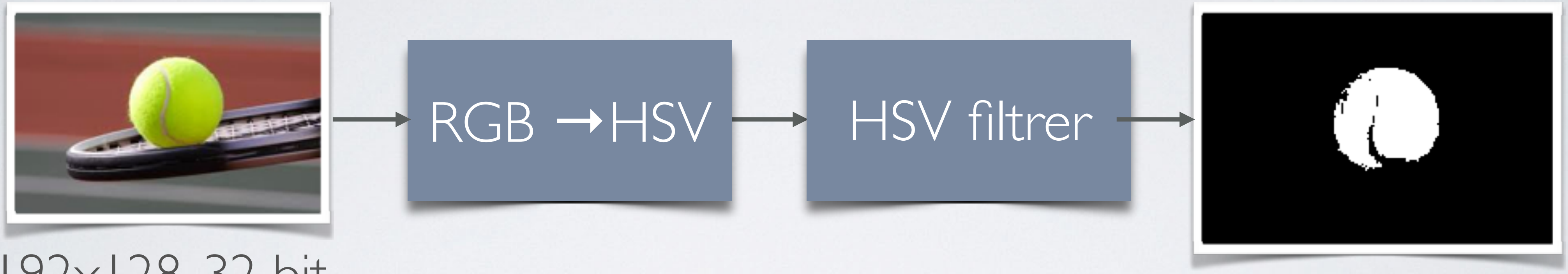
Object detector

Motors

Critical part !

🟦 Sensor ⬛ Smalltalk application 🟧 Actuators

PERFORMANCE COMPARISON



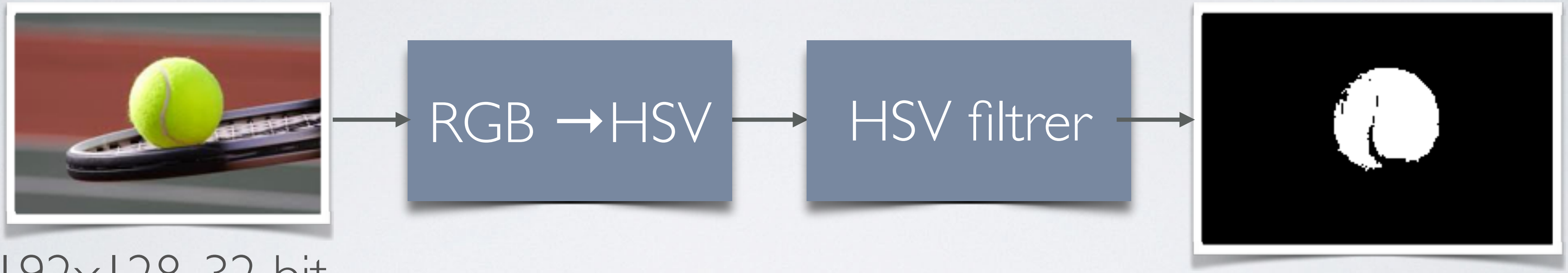
RGB → HSV

HSV filtrer

192x128, 32 bit

Pharo Smalltalk	C(OpenCV)	FPGA circuits
73 ms	**1.5ms**	**2.5ms**

PERFORMANCE COMPARISON

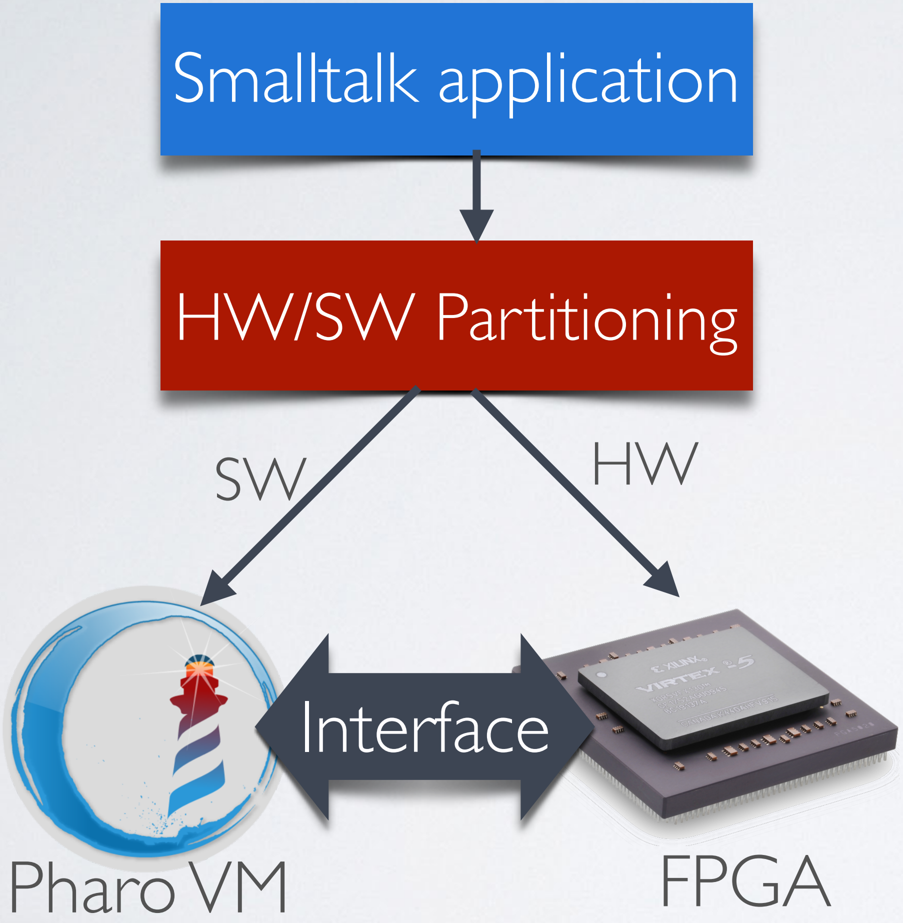


192x128, 32 bit

Pharo Smalltalk	C(OpenCV)	FPGA circuits
73 ms	**1.5ms**	**2.5ms**

! Bottleneck problem

CONCLUSION



- Future works :
  - Modelling methodology of hardware design using Smalltalk.
  - Pharo and FPGA interaction.
  - Software/hardware co-design integration.

TOWARD A METHODOLOGY TO TURN SMALLTALK CODE INTO FPGA

LE Xuan Sang[1,2]
Loïc LAGADEC[1], Luc FABRESSE[2],
Jannik LAVAL[2] and Noury BOURAQADI[2]

[1] Lab-STICC, ENSTA Bretagne
[2] Institut Mines-Telecom, Mines Douai