Make Life Easier for Embedded Software Engineers Facing Complex Hardware Architectures

R. Leconte, E. Jenn, G. Bois, H. Guérard

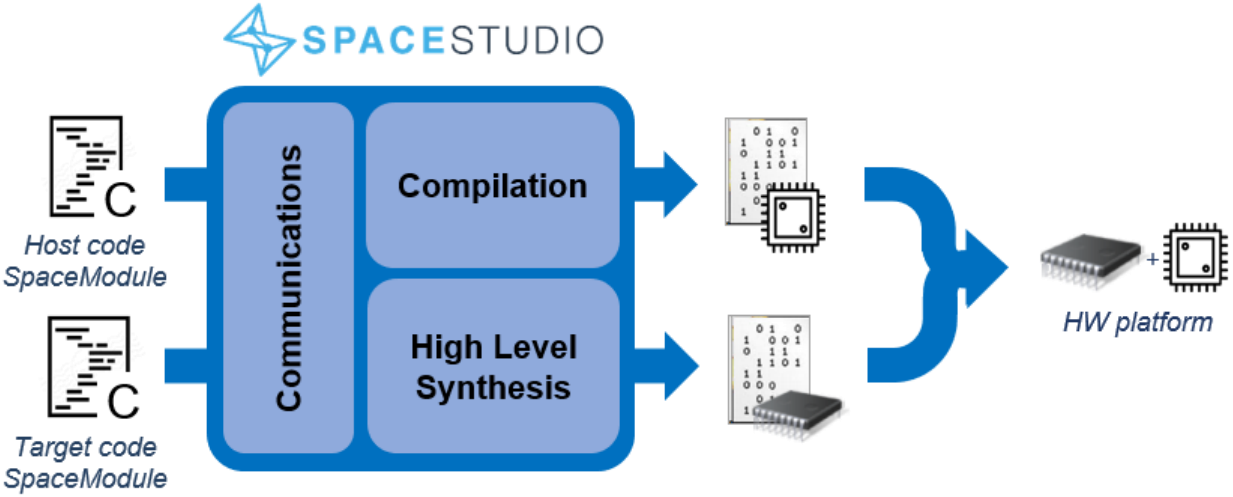
IRT Saint Exupéry, Space Codesign, Thales AVS

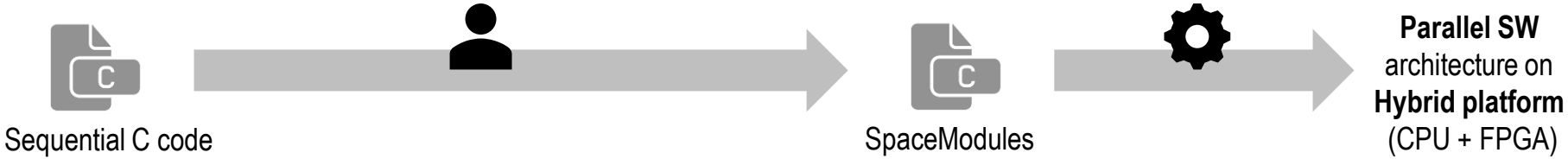
31/01/2020

AGENDA

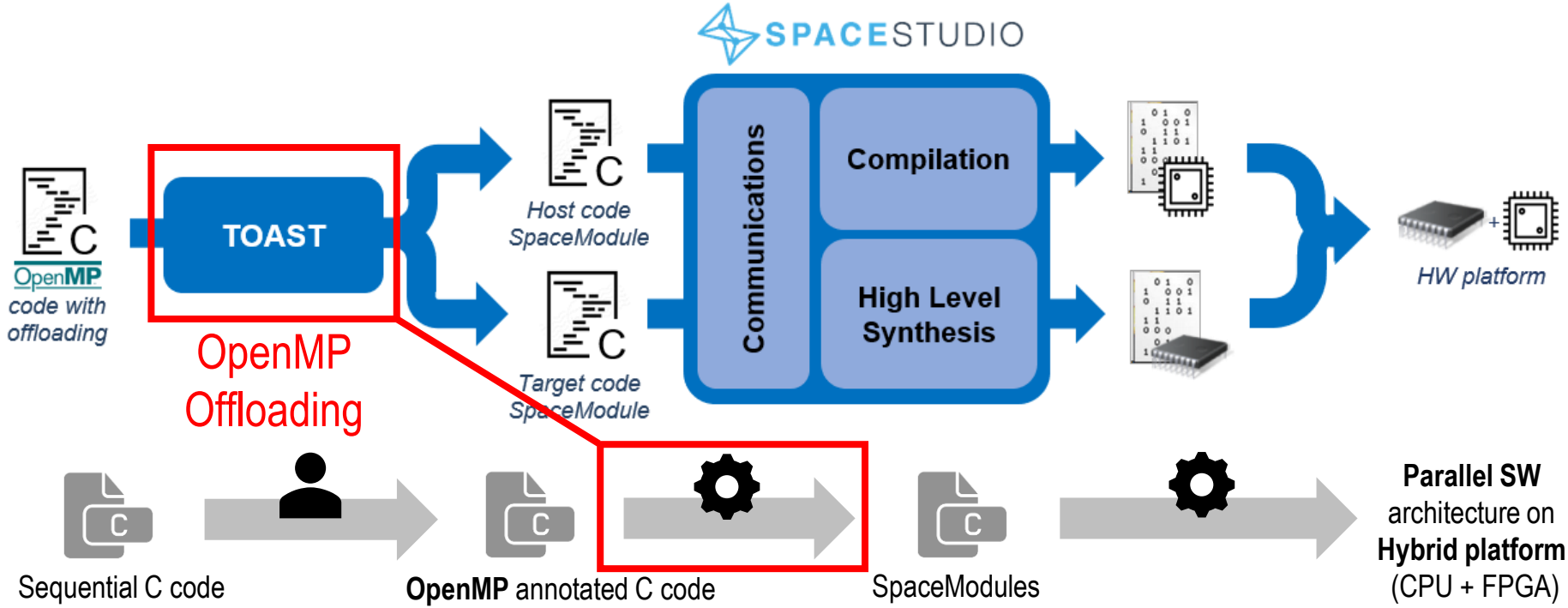
- SpaceStudio and deployment to hybrid HW/SW platform
- TOAST (Tool for OpenMP Annotations to Space design Translation) and its purpose
- OpenMP Offloading
- Example on a LineDetection algorithm

- Workflow based on an existing tool: **SpaceStudio** to **expose parallelism**
  - Encapsulate logic inside **SpaceModules**





- Use **OpenMP** to **expose parallelism**
  - Then **TOAST** generates **SpaceModules**
  - Finally **SpaceStudio deploys** to the **target platform**



- **OpenMP** allows to **offload** computations to an **accelerator:**
  - GPU, **FPGA**, etc.


```
int a = 2;
int b = 12;


            b = a + 2;

printf("b = %d\n", b);
// b = 4
```

Sequential C code

OpenMP annotated C code

SpaceModules

Parallel SW architecture on **Hybrid platform** (CPU + FPGA)

- **OpenMP** allows to **offload** computations to an **accelerator**
  - GPU, **FPGA**, etc.


```
int a = 2;
int b = 12;
#pragma omp target map(from:b)
{
            b = a + 2;
}
printf("b = %d\n", b);
// b = 4
```

Sequential C code

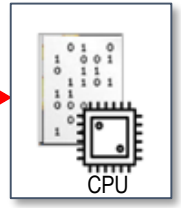
OpenMP annotated C code

SpaceModules

Parallel SW architecture on **Hybrid platform** (CPU + FPGA)

- **OpenMP** allows to **offload** computations to an **accelerator**
  - GPU, **FPGA**, etc.

```
int a = 2;
int b = 12;
#pragma omp target map(from:b)
{
        b = a + 2;
}
printf("b = %d\n", b);
// b = 4
```



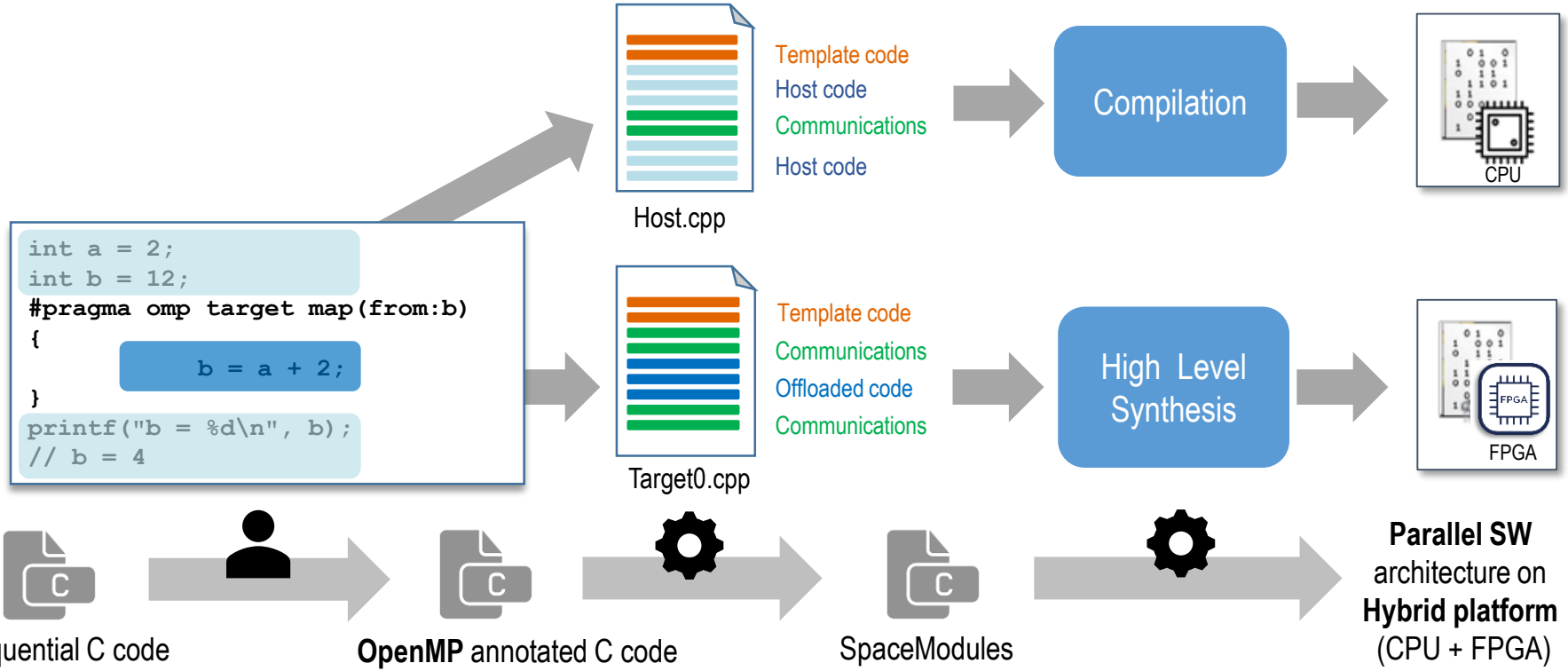
Sequential C code

OpenMP annotated C code

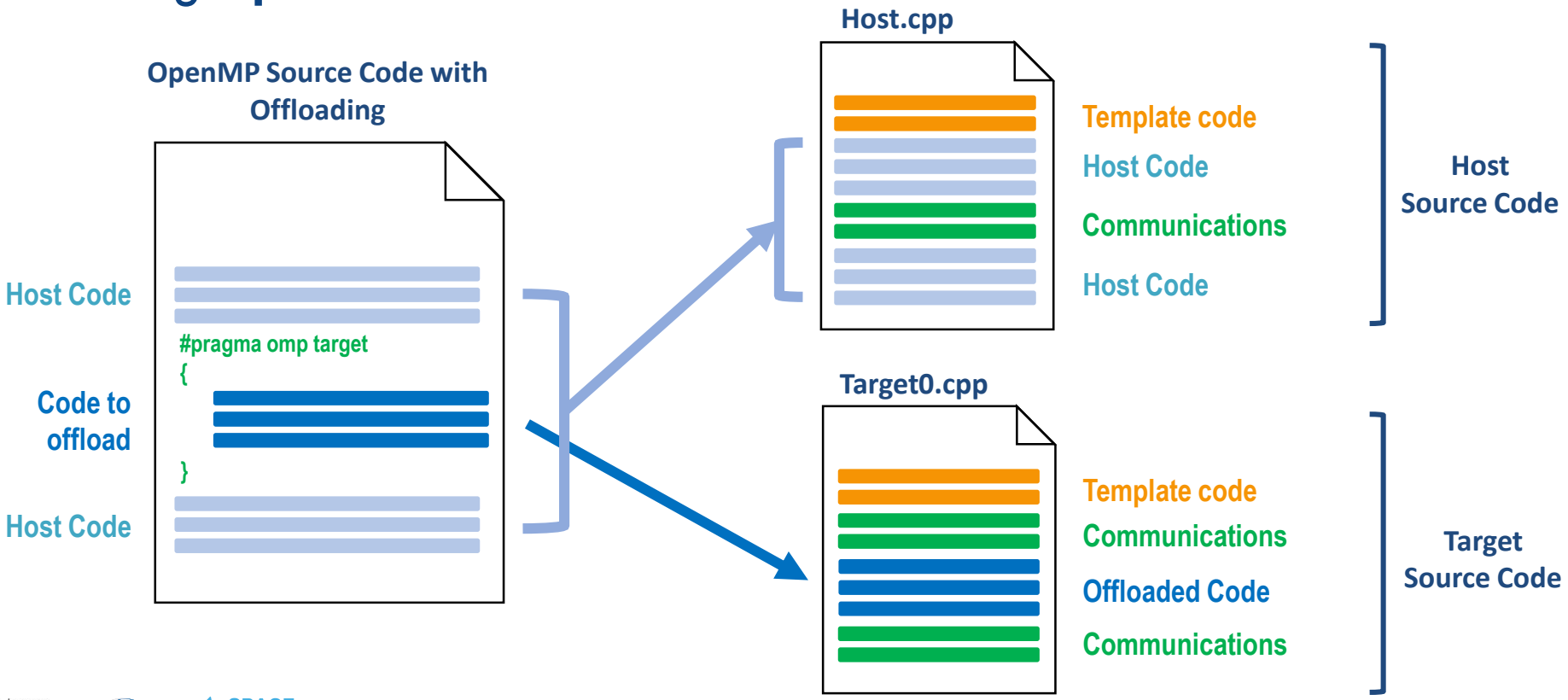
SpaceModules

Parallel SW architecture on **Hybrid platform** (CPU + FPGA)

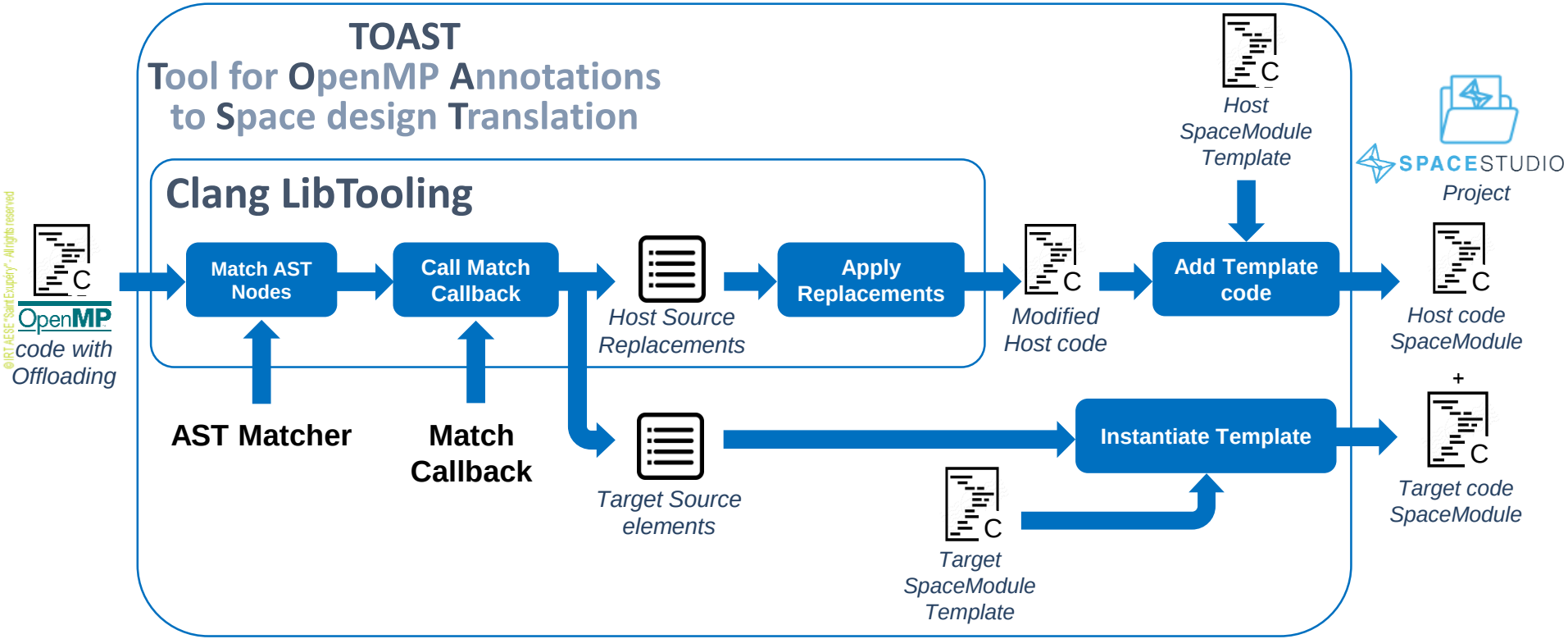
- **OpenMP** allows to **offload** computations to an **accelerator**
  - GPU, **FPGA**, etc.



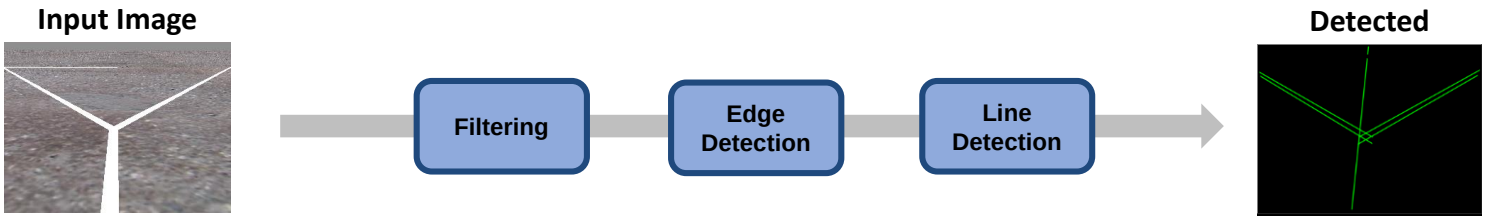
- **TOAST converts** OpenMP annotated code into C/C++ code using **SpaceStudio Communication APIs**



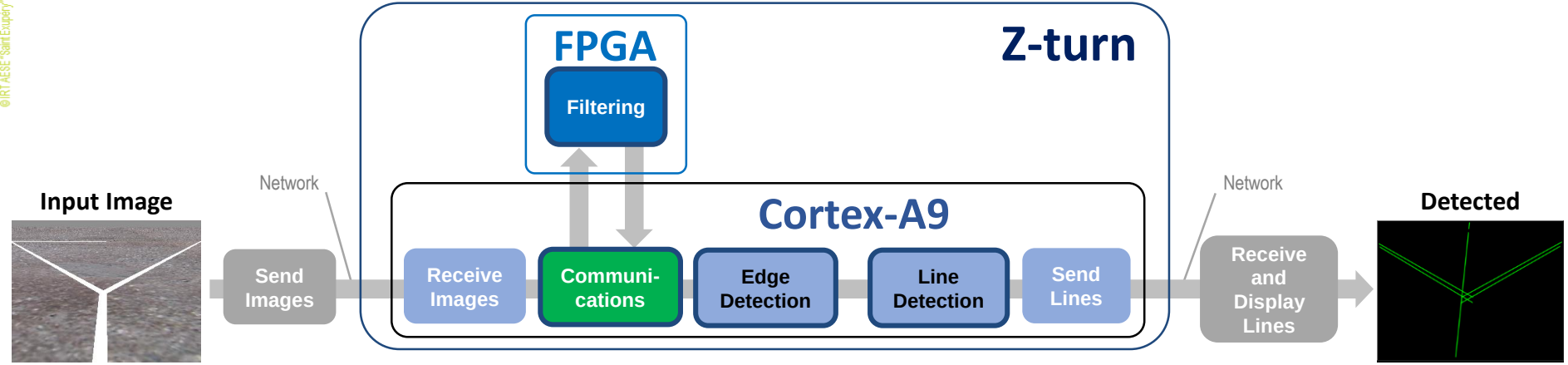
TOAST Architecture



LineDetection Use Case

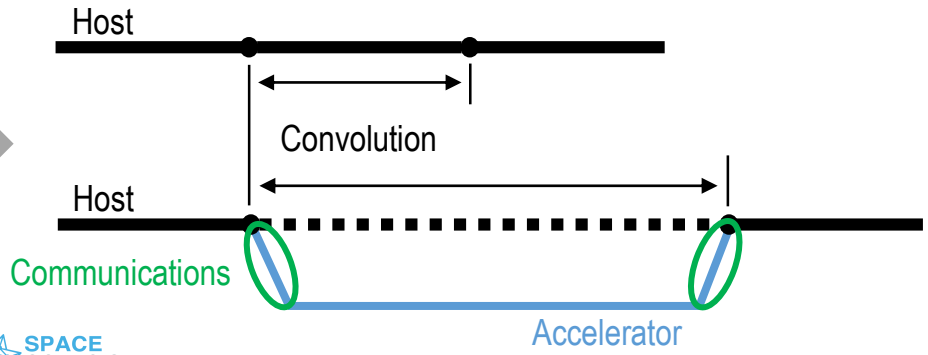


Line Detection and test bench



LineDetection: One Accelerator

```
#pragma omp declare target
void sub_filter(int line_start, int line_end, int height, int width, int kernel_size, unsigned
char* source_image, unsigned char* result_image) {
    // convolution accessing source_image and writing results in result_image
    ...
}
#pragma omp end declare target
...
#pragma omp target map(always, to: source_image[0:IMAGE_SIZE])
                   map(from: result_image[0:IMAGE_SIZE])
{
    sub_filter(0, height, height, width, kernel_size, source_image, result_image);
}
```



Slower because of :

- Communications
- Low FPGA frequency

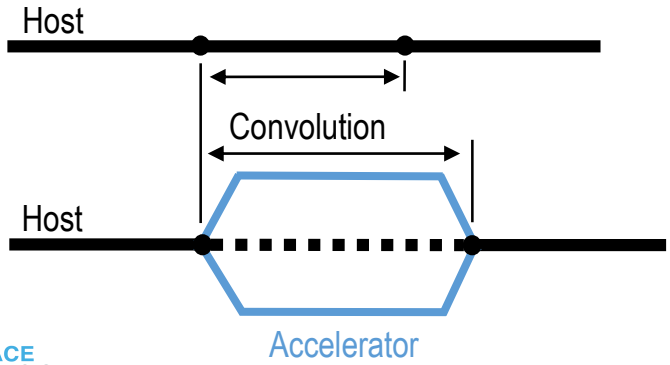
LineDetection : Several Accelerators

```
#pragma omp target map(always, to: source_image[0:IMAGE_SIZE])
                   map(from: result_image[0:IMAGE_SIZE])
{
    sub_filter(0, height, height, width, kernel_size, source_image, result_image);
}
```

```
#pragma omp target data map(always, to: source_image[0:IMAGE_SIZE])
                         map(from: result_image[0:IMAGE_SIZE])
{
#pragma omp target map(to: source_image[0: SLICE_SIZE + OVERLAP_SIZE])
                   map(from: result_image[0: SLICE_SIZE]) nowait
    { sub_filter(0, height / 2, height, width, kernel_size, source_image, result_image); }

#pragma omp target map(to: source_image[SLICE_SIZE - OVERLAP_SIZE: SLICE_SIZE + 2*OVERLAP_SIZE])
                   map(from: result_image[SLICE_SIZE:SLICE_SIZE]) nowait
    { sub_filter(height / 2, height, height, width, kernel_size, source_image, result_image); }
}
```

LineDetection : Several Accelerators

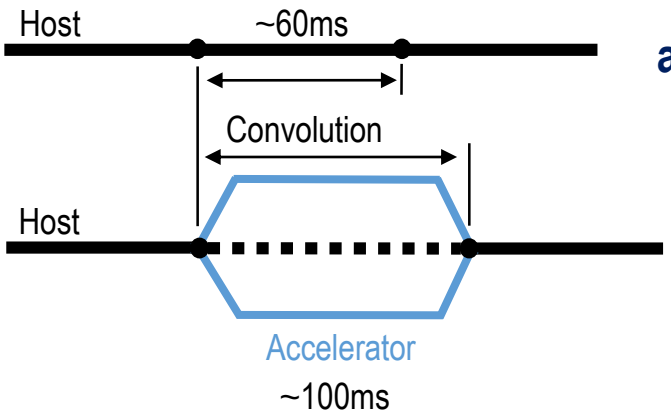
```
#pragma omp target data map(always, to: source_image[0:IMAGE_SIZE])
                        map(from: result_image[0:IMAGE_SIZE])
{
#pragma omp target map(to: source_image[0: SLICE_SIZE + OVERLAP_SIZE])
                   map(from: result_image[0: SLICE_SIZE]) nowait
    { sub_filter(0, height / 2, height, width, kernel_size, source_image, result_image); }

#pragma omp target map(to: source_image[SLICE_SIZE - OVERLAP_SIZE: SLICE_SIZE + 2*OVERLAP_SIZE])
                   map(from: result_image[SLICE_SIZE:SLICE_SIZE]) nowait
    { sub_filter(height / 2, height, height, width, kernel_size, source_image, result_image); }
}
```



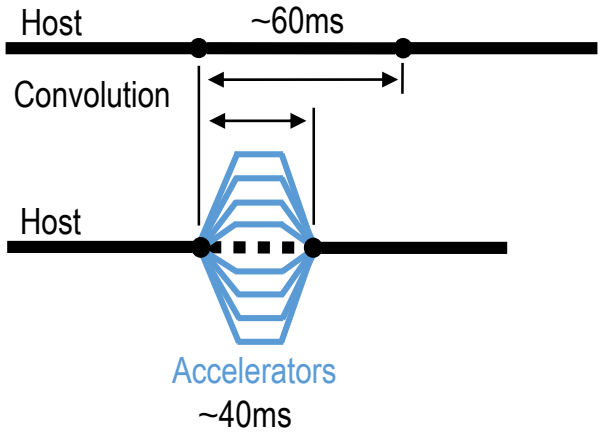
Evaluation of the solution :

- **Execution time**
- **Space occupation** on the **FPGA**

LineDetection : Several Accelerators



Small code changes for
architecture exploration

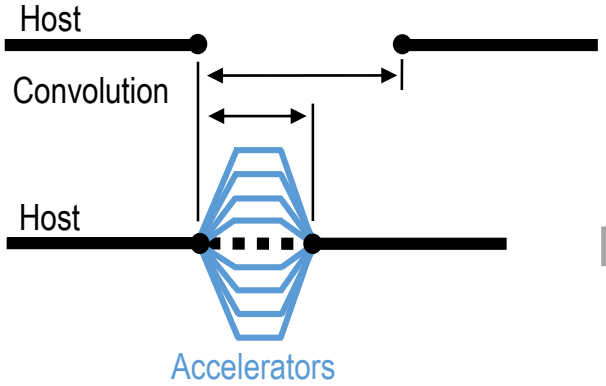


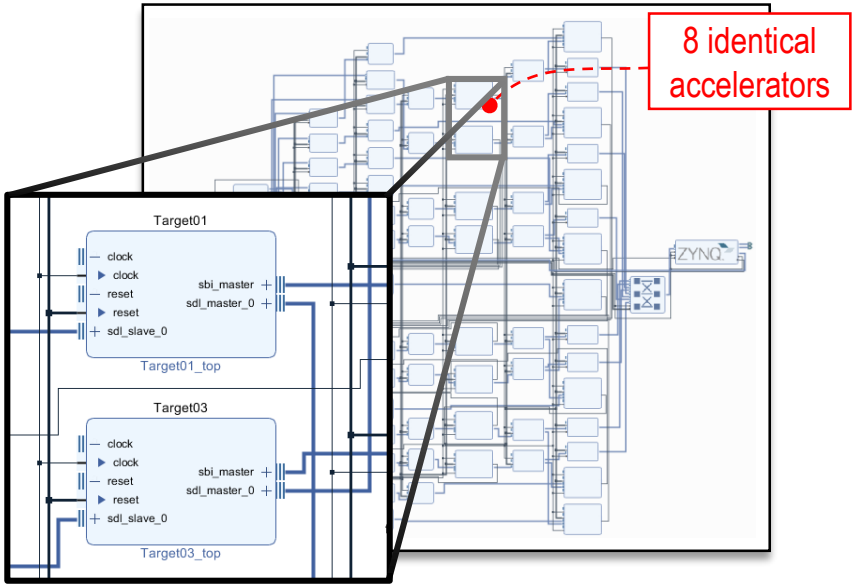
Possible additional work:

- **HLS pragmas -> ~17ms**

- **OpenMP** allows to **offload** computations to **multiple accelerators**
  - **Architecture exploration:**

    allows to find the good offloading pattern





Sequential C code

OpenMP annotated C code

Parallel SW architecture on **Hybrid platform** (CPU + FPGA)

Architectures in SpaceStudio

Xilinx : Zynq

- LineDetection
  - OpenMP
    - User components
    - ApplicationDefinitions.h
    - functional
    - zynq (active)
      - AMBA_AXIBus_LT0
        - zynq0
          - Host0
          - build.ninja
          - main.cpp
        - XilinxPIC0
      - zynq0_HP0_AMBA_AXIBus_LT0
        - Target00
        - Target10
        - Target20
        - Target30
        - Target40
        - Target50
        - Target60
        - Target70
        - zynq0_DDR_InterfaceRange0
      - reset_manager
      - SysClock
      - PlatformDefinitions.h
      - main.cpp
      - main.h
      - bus_mapping.cpp
      - bus_mapping.h
      - build.ninja
    - cyclone_v_soc

Intel : Cyclone

- LineDetection
  - OpenMP
    - User components
    - ApplicationDefinitions.h
    - functional
    - zynq
    - cyclone_v_soc (active)
      - AMBA_AXIBus_LT0
        - cyclone_v_soc0
          - Host0
          - build.ninja
          - main.cpp
      - cyclone_v_soc0_F2H_AMBA_AXIBus_LT0
        - Target00
        - Target10
        - Target20
        - Target30
        - Target40
        - Target50
        - Target60
        - Target70
        - cyclone_v_soc0_DDR_InterfaceRange0
      - reset_manager
      - SysClock
      - PlatformDefinitions.h
      - main.cpp
      - main.h
      - bus_mapping.cpp
      - bus_mapping.h
      - build.ninja

SpaceStudio 3.3.9 (56 days left) - C:/LineDetection_projects/generated_8_acc_nowait

File Edit Solution Tools Run Window Help

Generate...
Clean generated files
Build
FPGA Resource Estimation...
Architecture Implementation...
Preferences...

Project Explorer

- LineDetection
  - OpenMP
  - User com
  - Applicati
  - functiona
  - zynq (ac
    - AMBA_AXIBus_LT0
      - zynq0
        - Host0
        - build.ninja
        - main.cpp
      - XilinxPIC0
    - zynq0_HP0_AMBA_AXIBus_LT0
      - Target00
      - Target10
      - Target20
      - Target30
      - Target40
      - Target50
      - Target60
      - Target70
      - zynq0_DDR_InterfaceRange0
    - reset_manager
    - SysClock
    - PlatformDefinitions.h
    - main.cpp
    - main.h
    - bus_mapping.cpp
    - bus_mapping.h
    - build.ninja
  - cyclone_v_soc

ild | Output | Host.h | Host.cpp | Target1.h | Target1.cpp

utput

```
Generation of 'main.cpp' ...  [Successful.]

Generation of 'main.h' ...  [Successful.]

Generation of 'build.ninja' ...  [Successful.]

Generation of 'JSON files' ...  [Successful.]

Generation of 'communication.cpp' ...  [Successful.]

Generation of 'application_os_requirement.h' ...  [Successful.]

Generation of 'application_os_requirement.cpp' ...  [Successful.]

Generation of 'main.cpp' ...  [Successful.]

Generation of 'build.ninja' ...  [Successful.]

Generation of 'load_drivers_script.sh' ...  [Successful.]

Generation of 'ip_saver.cpp' ...  [Successful.]

Generation of 'gdbserver_notifier.cpp' ...  [Successful.]

Generation of 'profiling_saver.cpp' ...  [Successful.]
```

SpaceStudio 3.3.9 (56 days left) - C:/LineDetection_projects/generated_8_acc_nowait

File Edit Solution Tools Run Window Help

Project Explorer

- LineDetection
  - OpenMP
    - User components
    - ApplicationDefinitions.h
    - functional
    - zynq (active)
      - AMBA_AXIBus_LT0
        - zynq0
          - Host0
          - build.ninja
          - main.cpp
        - XilinxPIC0
      - zynq0_HP0_AMBA_AXIBus_LT0
        - Target00
        - Target10
        - Target20
        - Target30
        - Target40
        - Target50
        - Target60
        - Target70
        - zynq0_DDR_InterfaceRange0
      - reset_manager
      - SysClock
      - PlatformDefinitions.h
      - main.cpp
      - main.h
      - bus_mapping.cpp
      - bus_mapping.h
      - build.ninja
    - cyclone_v_soc

Build | Output | Host.h | Host.cpp | Target1.h | Target1.cpp

Build output

```
Generation of 'main.cpp' ... [Successful.]
```

Architecture Implementation export

Architecture Implementation

Implements a complete system on a physical FPGA board

Project directory: C:/implementation/zturn_8_acc [Browse...]

Electronic Design Automation (EDA) tool: Xilinx - Vivado 2018.3

Board: Z-turn 7z020

High-level synthesis: Vivado HLS

Modules to synthesize

Module name
☑ Target00
☑ Target10
☑ Target20
☑ Target30
☑ Target40
☑ Target50
☑ Target60
☑ Target70

OK Cancel

```
Generation of 'profiling_saver.cpp' ... [Successful.]
```

SpaceStudio 3.3.9 (56 days left) - C:/LineDetection_projects/generated_8_acc_nowait

File Edit Solution Tools Run Window Help

Project Explorer

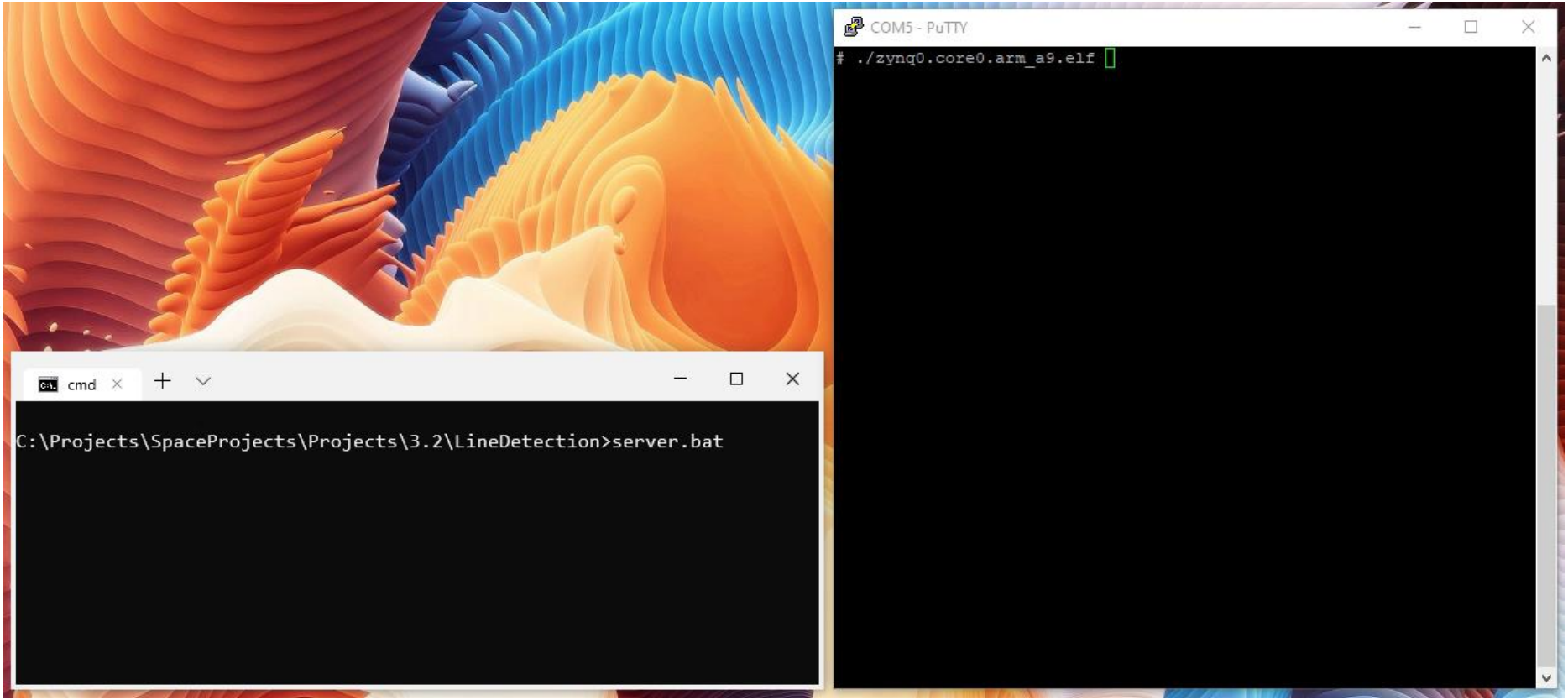
- LineDetection
  - OpenMP
    - User components
    - ApplicationDefinitions.h
    - functional
    - zynq (active)
      - AMBA_AXIBus_LT0
        - zynq0
          - Host0
          - build.ninja
          - main.cpp
        - XilinxPIC0
      - zynq0_HP0_AMBA_AXIBus_LT0
        - Target00
        - Target10
        - Target20
        - Target30
        - Target40
        - Target50
        - Target60
        - Target70
        - zynq0_DDR_InterfaceRange0
      - reset_manager
      - SysClock
      - PlatformDefinitions.h
      - main.cpp
      - main.h
      - bus_mapping.cpp
      - bus_mapping.h
      - build.ninja
    - cyclone_v_soc

Build | Output | Host.h | Host.cpp | Target1.h | Target1.cpp

Build output

```
'Finished building: C:/implementation/zturn_8_acc_network/zynq.sdk/zynq/ps7_init.c'
' '
'Building file: ../src/qspi.c'
'Invoking: ARM v7 gcc compiler'
arm-none-eabi-gcc -Wall -O0 -g3 -I"C:\implementation\zturn_8_acc_network\zynq.sdk\zynq" -c -
'Finished building: ../src/qspi.c'
' '
'Building file: ../src/rsa.c'
'Invoking: ARM v7 gcc compiler'
arm-none-eabi-gcc -Wall -O0 -g3 -I"C:\implementation\zturn_8_acc_network\zynq.sdk\zynq" -c -
'Finished building: ../src/rsa.c'
' '
'Building file: ../src/sd.c'
'Invoking: ARM v7 gcc compiler'
arm-none-eabi-gcc -Wall -O0 -g3 -I"C:\implementation\zturn_8_acc_network\zynq.sdk\zynq" -c -
'Finished building: ../src/sd.c'
' '
'Building target: fsbl.elf'
'Invoking: ARM v7 gcc linker'
arm-none-eabi-gcc -mcpu=cortex-a9 -mfpu=vfpv3 -mfloat-abi=hard -Wl,-build-id=none -specs=Xil
'Finished building target: fsbl.elf'
' '
'Invoking: ARM v7 Print Size'
arm-none-eabi-size fsbl.elf  |tee "fsbl.elf.size"
   text    data     bss     dec     hex filename
  81496   11928   74712  168136   290c8 fsbl.elf
'Finished building: fsbl.elf.size'
' '

11:39:45 Build Finished (took 4s.358ms)

Invoking scanner config builder on project
Building '/zynq'

Project Implementation [Completed successfully.]
```

Video: application running on Z-turn



Conclusion

TOAST allows to leverage the **OpenMP Offloading** standard to deploy software on **FPGA**s using **SpaceStudio**

- OpenMP is a well established standard

Our workflows allows:

- Design space exploration
- Deployment on SoCs featuring an FPGA
- Support multiple vendors (Xilinx, Intel)

Thank you for your attention