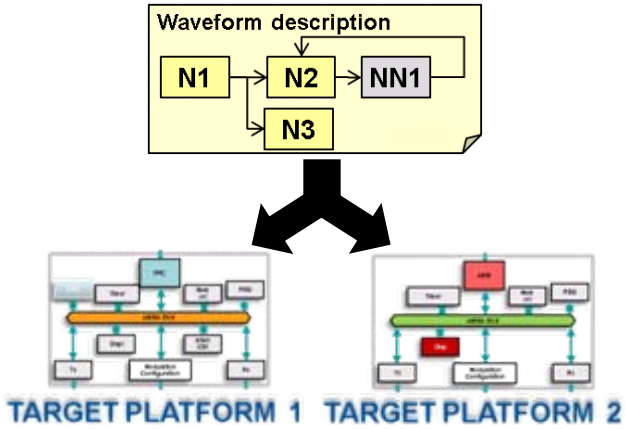
Component-based Waveform Development: The Nucleus Tool Flow for Efficient and Portable SDR

J. Castrillon, S. Schürmans, A. Stulova, W. Sheng, T. Kempf, A. Ishaque,
R. Leupers, G. Ascheid, H. Meyr
Jeronimo.Castrillon@iss.rwth-aachen.de

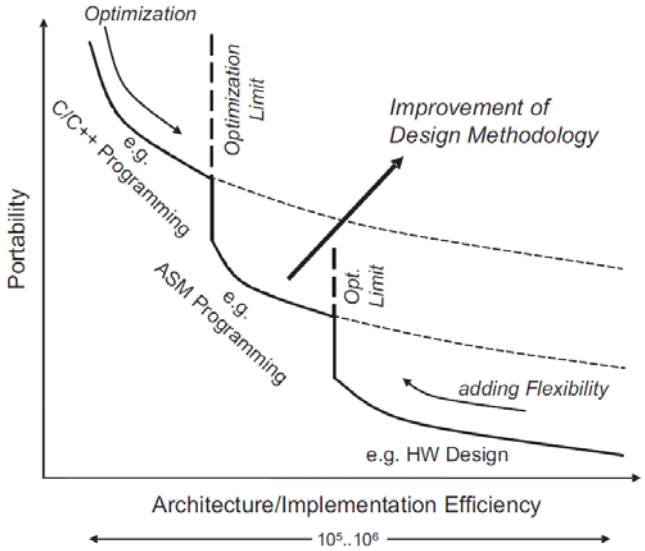
SDR Dec. 2010

Institute for Communication Technologies and Embedded Systems

- **Benefits of Component-based SW Engineering (CBSE)**
  - SW productivity increase
  - Familiar block diagram paradigm
  - Potentially portable specification

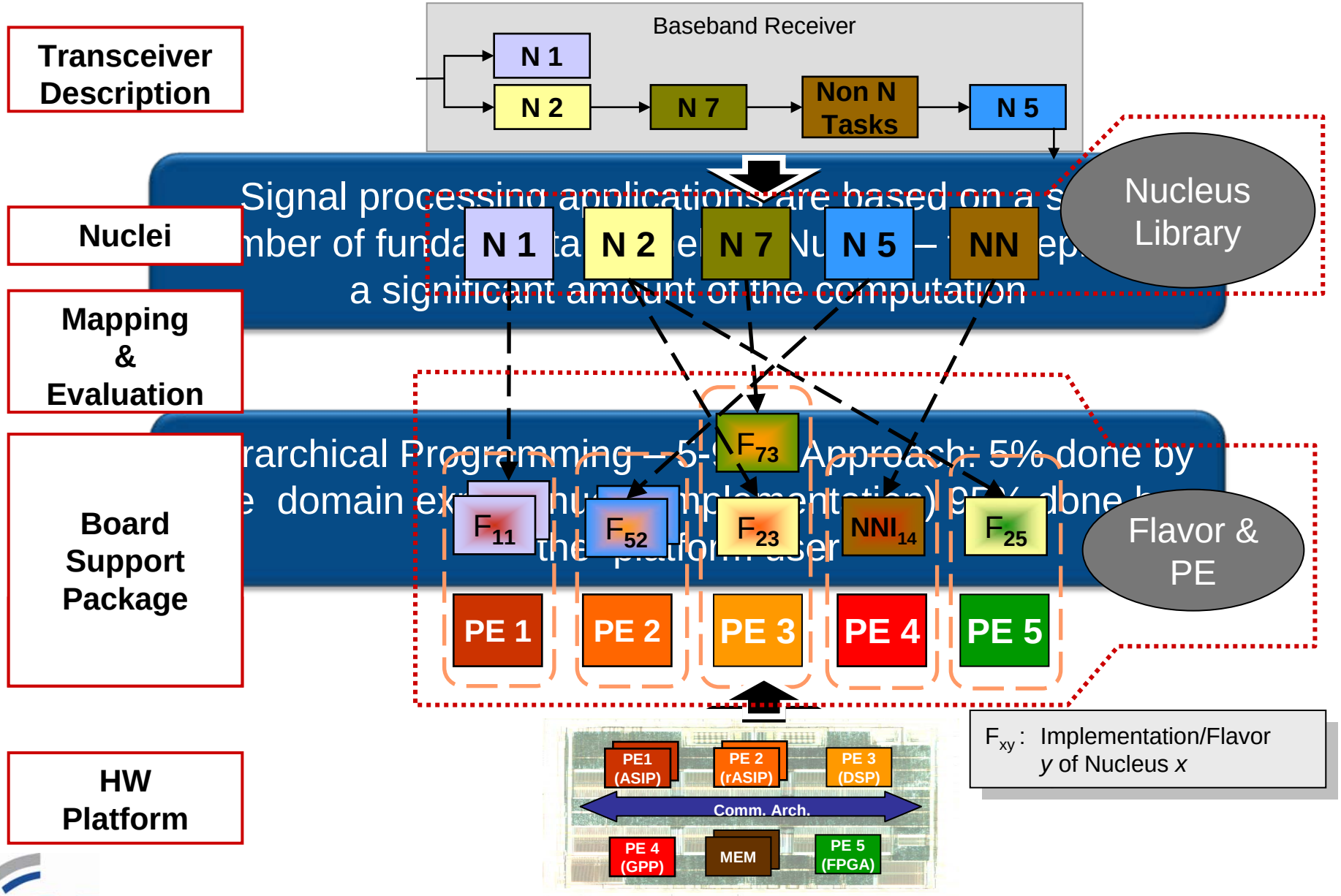


- **Problems:**
  - Portability vs. efficiency
    - Big gap between high level languages and architectures
  - Efficiency: normally achieved with the library approach
    - Architecture dependent



Source: Witte et al. WSR'08

Nucleus Approach: Overview

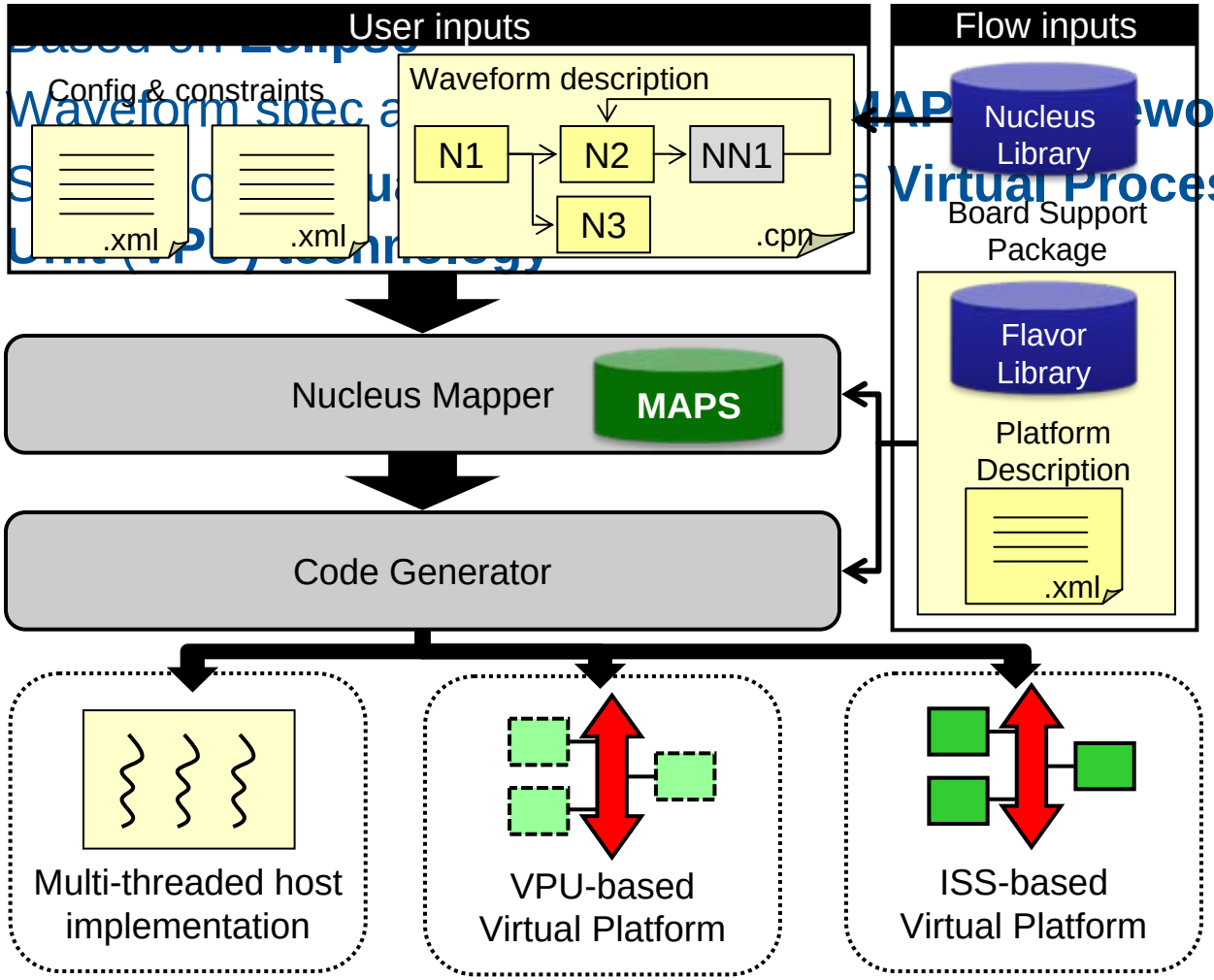


Outline

- **Motivation**
- **Tool Flow:**
  - Overview
  - Inputs
  - Waveform mapping
  - Code generation
- **Case study**
- **Summary and Outlook**

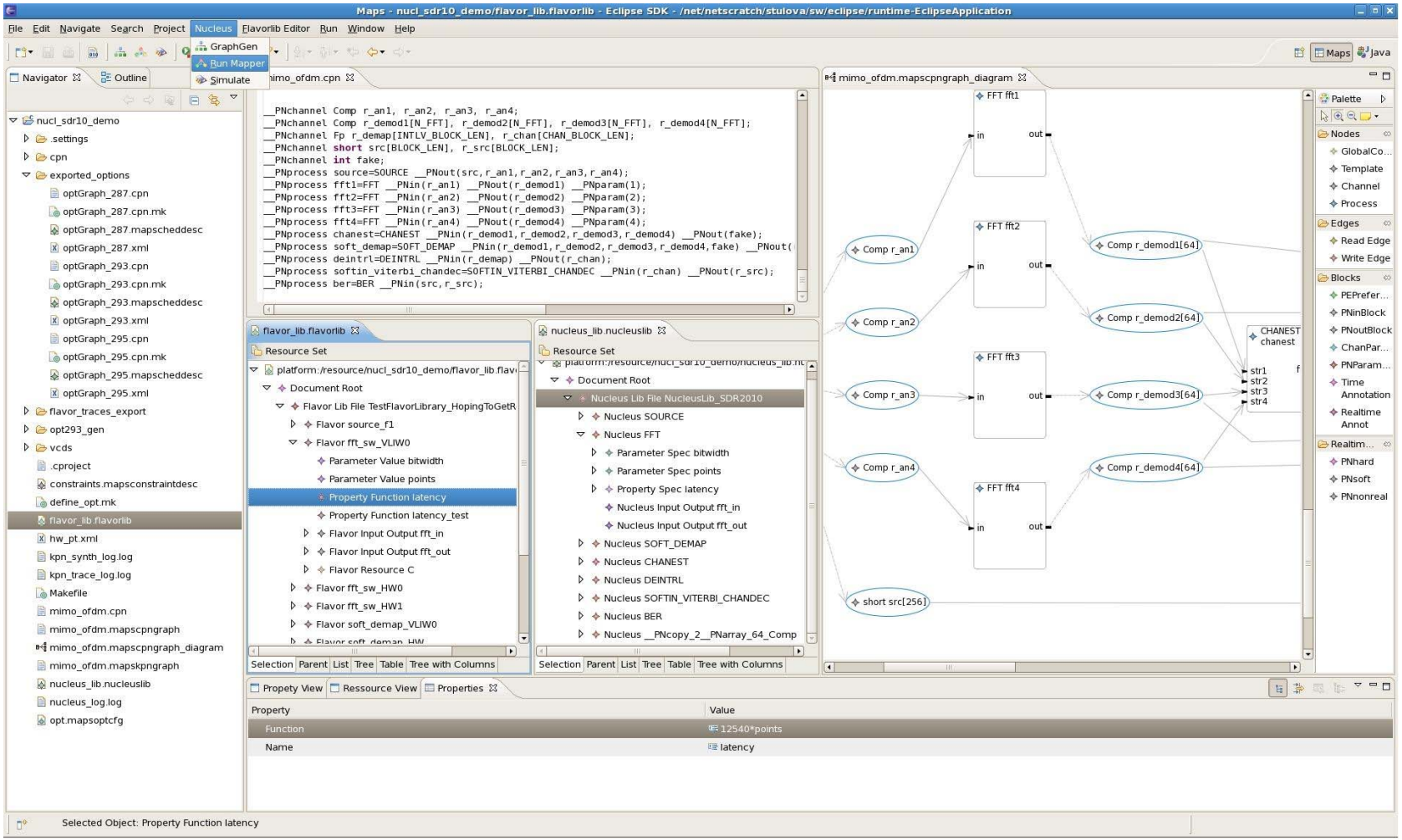
Tool Flow: Overview

- **Waveform Development Environment (WDE)**
  - **Based on Eclipse**
  - **Waveform spec a… …MAPS framework**
  - **S… d… u… e Virtual Processing Unit (VPU) technology**

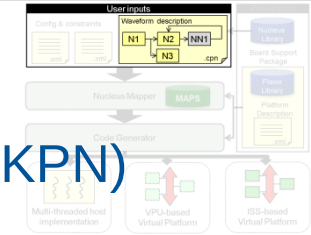


Tool Flow: Overview (2)

- **GUI:**



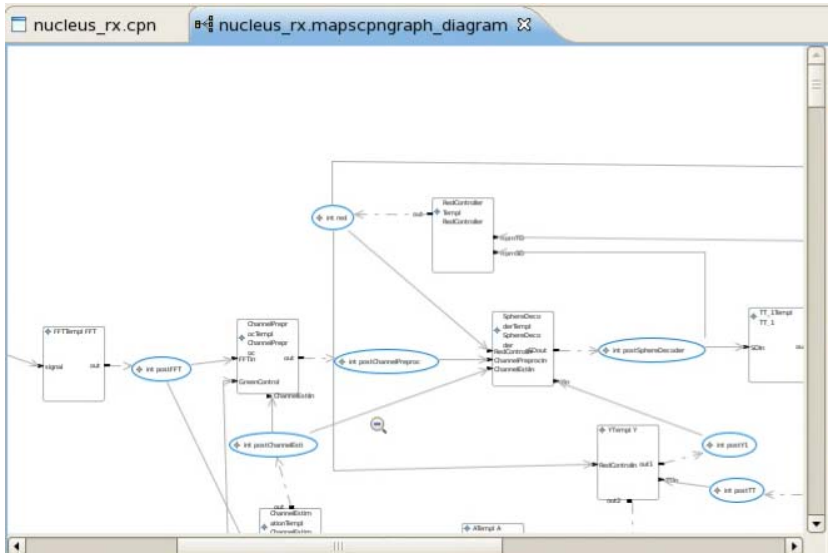
Tool Flow: Inputs



- **Waveform description: MAPS CPN Language**
  - General dataflow model: Kahn Process Networks (KPN)
  - Small extension to C (few new keywords)

```
__PNkpn myproc __PNin(int A)
__PNout(char B[3][3])
{
  int i, j;
  __PNout(B)
    for (i = 0; i < 3; i++)
      for (j = 0; j < 3; j++)
        B[i][j] = 5;
  while (1)
    __PNin(A) __PNout(B)
      for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
          B[i][j] = (char)A;
}
...
```

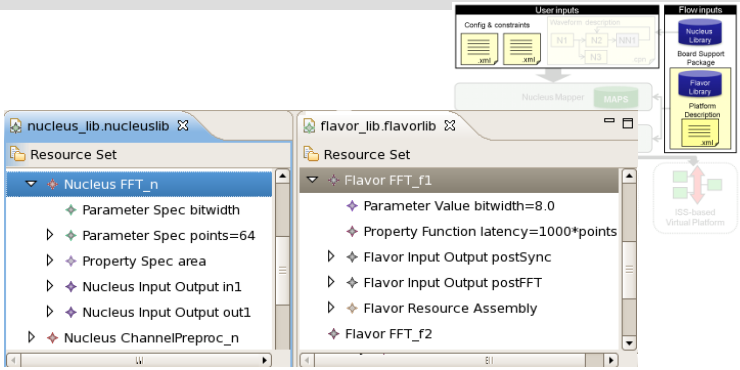
.cpn



.vcpn

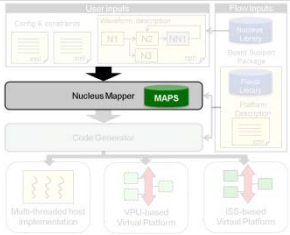
Tool Flow: Inputs (2)

- **Libraries for Nuclei and Flavors**
  - XML representation
  - Efforts in finding common parameters and properties

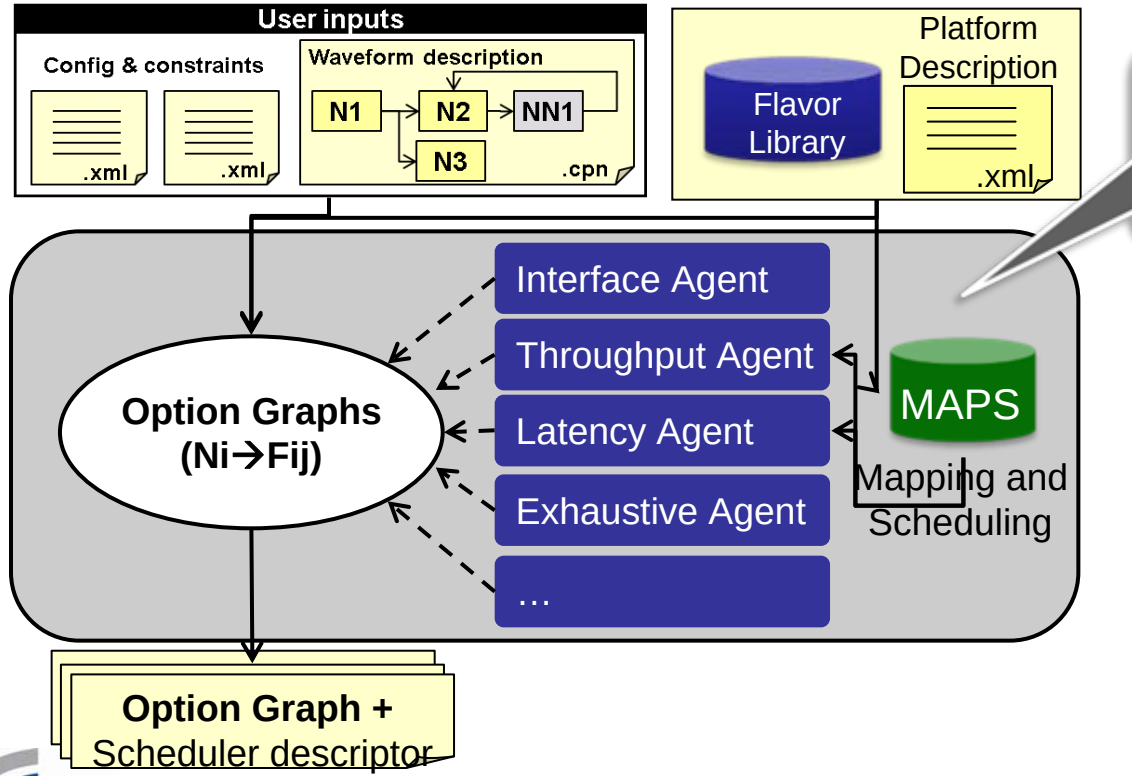


- **Constraints:**
  - XML representation
  - Supported: latency along paths, throughput at different locations, multi-tasking

- **Configuration:**
  - XML representation
  - Allow to configure several parameters of the tool

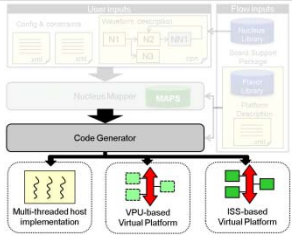


- **Mapping: Finding a flavor for every nucleus**
  - Architecture: list of *Option-Graphs* that is refined by running *Agents*
  - Export feasible options for further analysis

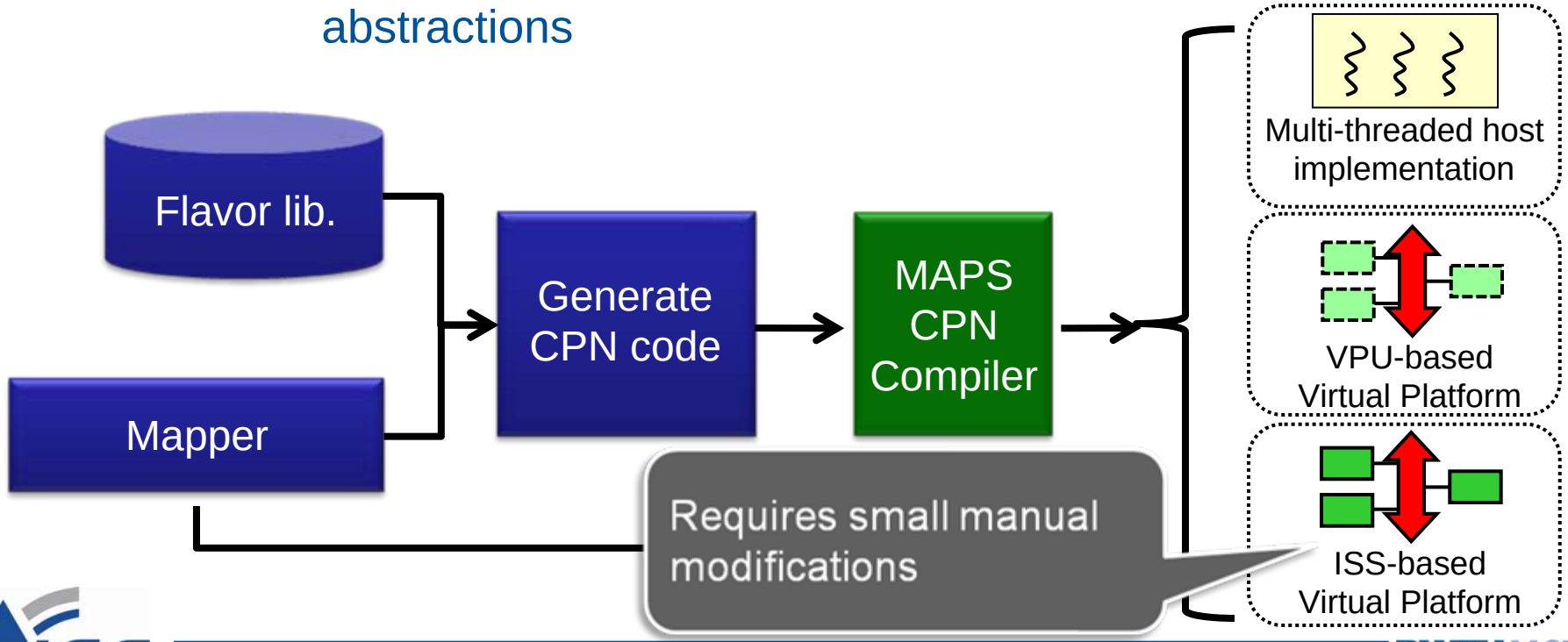



```
// an ordered list of smart-agents
Mapper.regPass(interface);
Mapper.regPass(critical_path);
Mapper.regPass(throughput);
...
```

Tool Flow: Code Generation



- **Option graphs include:**
  - A mapping of nuclei to flavors
  - A scheduler descriptor
- **Code-gen and simulation**
  - Pick the right implementation from the Flavor library
  - Export simulation with scheduling information: different abstractions



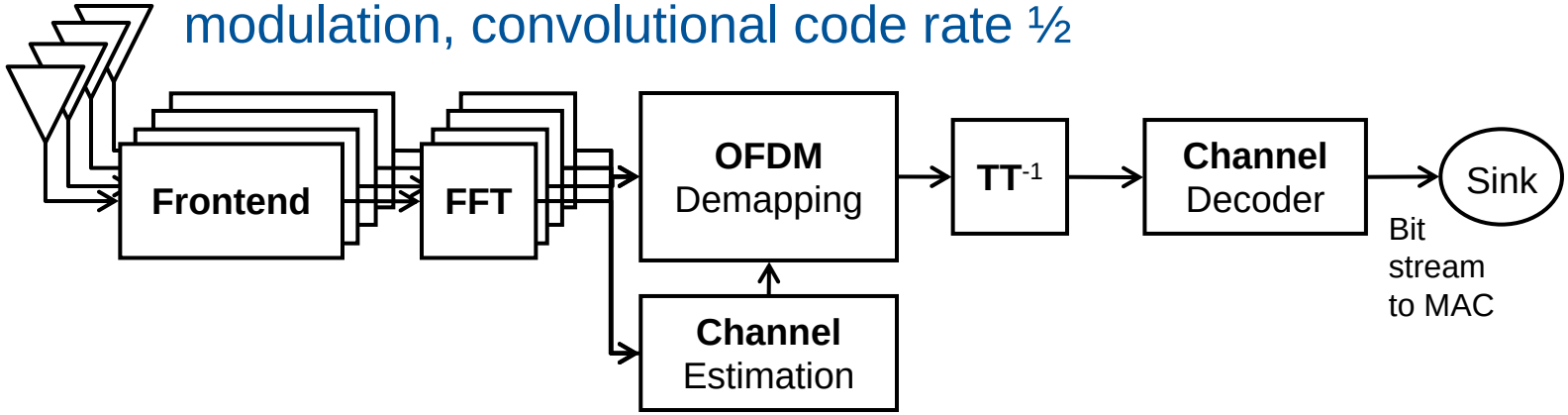
Outline

- **Motivation**
- **Tool Flow:**
  - Overview
  - Inputs
  - Waveform mapping
  - Code generation

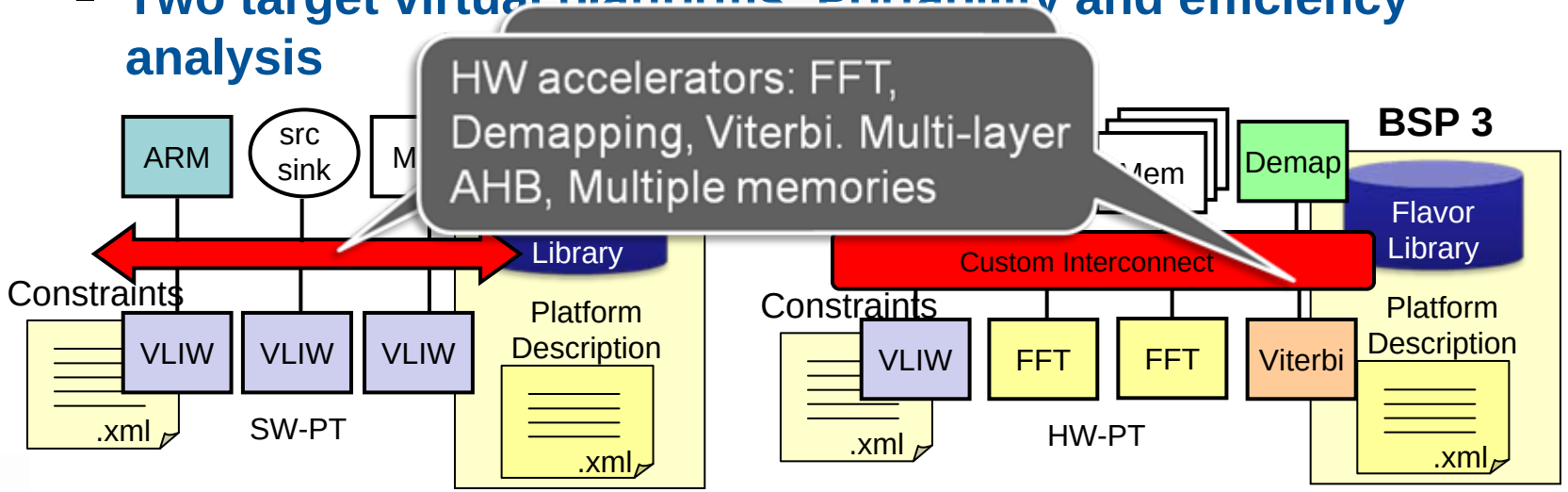
Case study

- **Summary and Outlook**

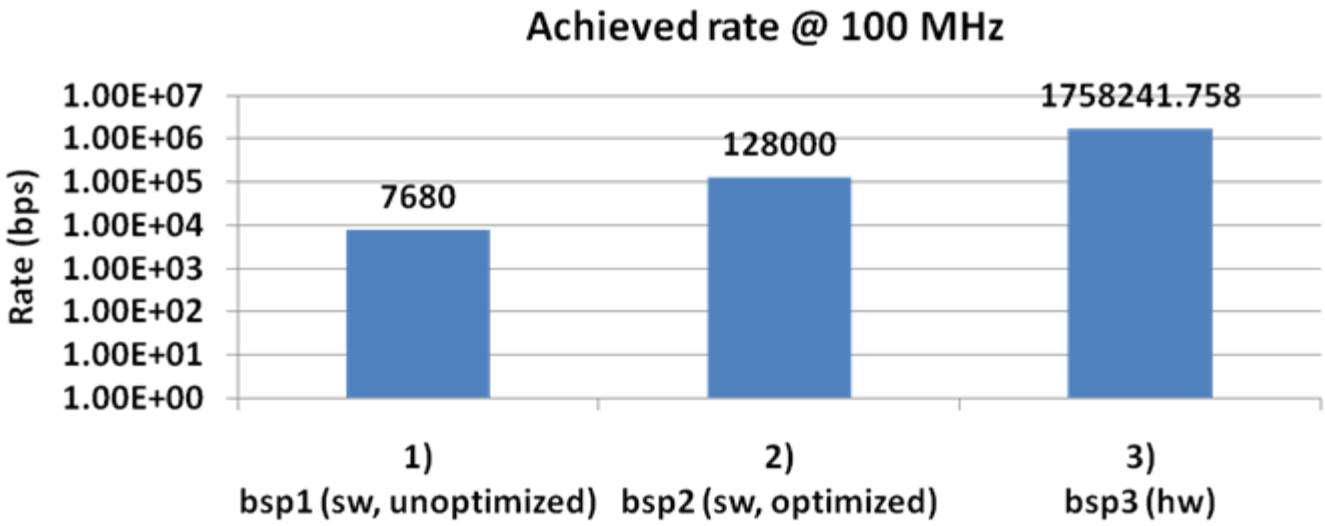
- **Application: MIMO OFDM Receiver**
  - 4 Antennas, 64 sub-carriers (48 with payload), QPSK modulation, convolutional code rate ½



- **Two target virtual platforms: Portability and efficiency analysis**



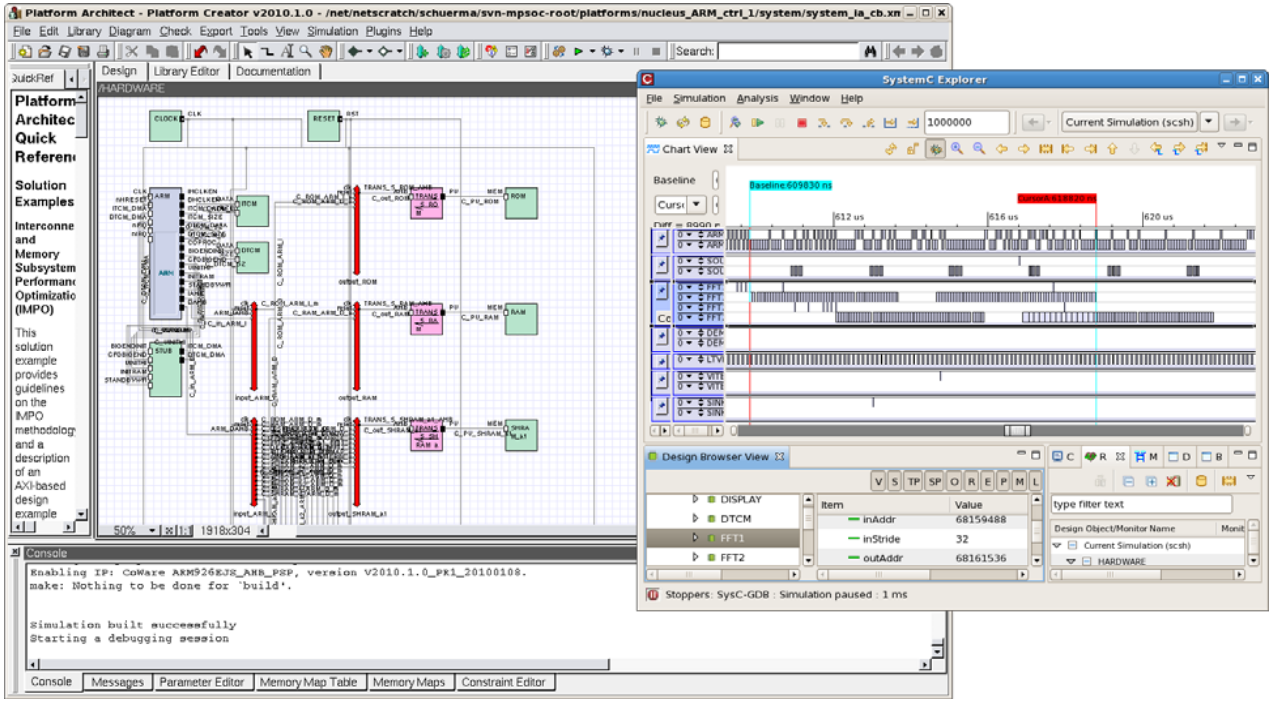
- **Best solutions found by the tool for each BSP with the same application specification (w/o simulation)**



- **Tool performance:**
  - Option-graphs analysis: from **~600** to **3** options in **~15s** (on a dual core AMD Opteron, 2.6 GHz.)

Case Study: Results (2)

- **Exporting simulations:**



  - Actual rate on cycle accurate platform 16 % slower than predicted

Outline

- **Motivation**
- **Tool Flow:**
  - Overview
  - Inputs
  - Waveform mapping
  - Code generation
- **Case study**

Summary and Outlook

Summary and Outlook

- **Summary:**
  - A flow from KPN waveform description to implementation
  - Portability: high level description based on components (***nuclei***)
  - Efficiency: tool selects efficient implementations (***flavors***) from BSP
  - Integrated with state-of-the-art ESL tools

- **Outlook:**
  - Characterization of new nuclei
  - Implementation and characterization of new flavors
  - Analysis of iterative decoding

Thanks!

Questions?

ACK: This work has been supported by the UMIC (Ultra High-Speed Mobile Information and Communication) research centre. www.umic.rwth-aachen.de

Contact: Jeronimo.Castrillon@iss.rwth-aachen.de