# NumPEx

*High Performance Numerics for Exascale*

# NumPEx / ExaSoft

*Exascale computing*

## High Performance Computing software and tools

### Challenges:

- *Harness the power of highly accelerated, large scale architectures*
- *Support portable functionality and performance across different architectures*
- *Provide better separation of concerns between algorithms and implementations*

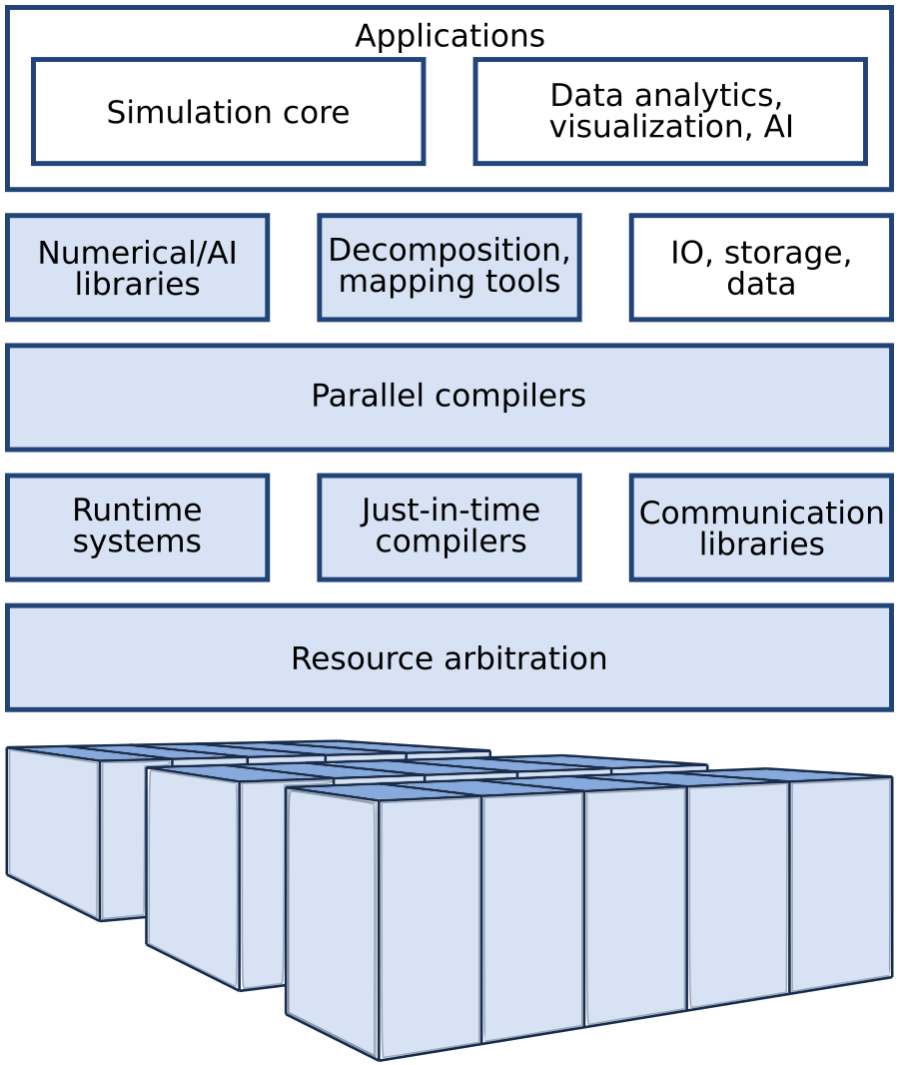
### Objectives:

- *Develop programming models, runtime systems and compilers to achieve high productivity and portable performance on large-scale heterogeneous systems*
- *Develop new performance and energy profiling and optimization approaches and tools*
- *Produce a new generation of scalable, portable and composable numerical libraries*

### Team:

*Inria, CNRS, CEA, UPSaclay, Telecom SudParis*

## The NumPEx software stack



# NumPEx / ExaSoft

*Exascale computing*

## High Performance Computing software and tools

### Challenges:

- *Harness the power of highly accelerated, large scale architectures*
- *Support portable functionality and performance across different architectures*
- *Provide better separation of concerns between algorithms and implementations*

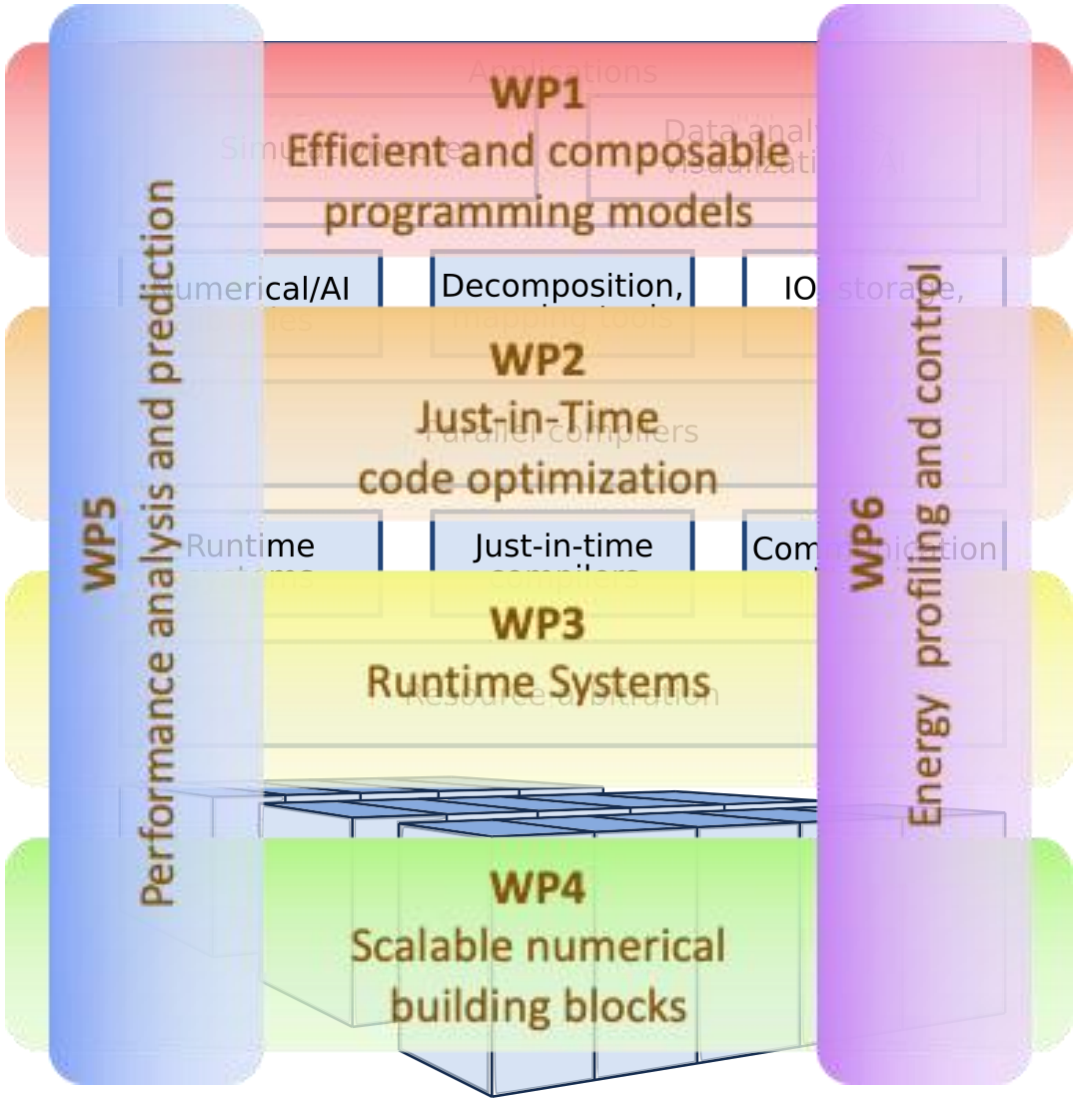
### Objectives:

- *Develop programming models, runtime systems and compilers to achieve high productivity and portable performance on large-scale heterogeneous systems*
- *Develop new performance and energy profiling and optimization approaches and tools*
- *Produce a new generation of scalable, portable and composable numerical libraries*

### Team:

*Inria, CNRS, CEA, UPSaclay, Telecom SudParis*

The NumPEx software stack



# NumPEx / ExaSoft

*Exascale computing*

## WP1: Efficient and composable programming models

### Objectives:

- *Abstract the application (code and data) from the machines*
  - *Ease the porting to new machines*
  - *Ease the exploitation of machine heterogeneity*
  - *Ease debugging of complex software stack*
- *Improve code composability*
  - *Simplify code reuse to reduce the development*
  - *Improve the separation of concerns between domain experts and HPC experts*

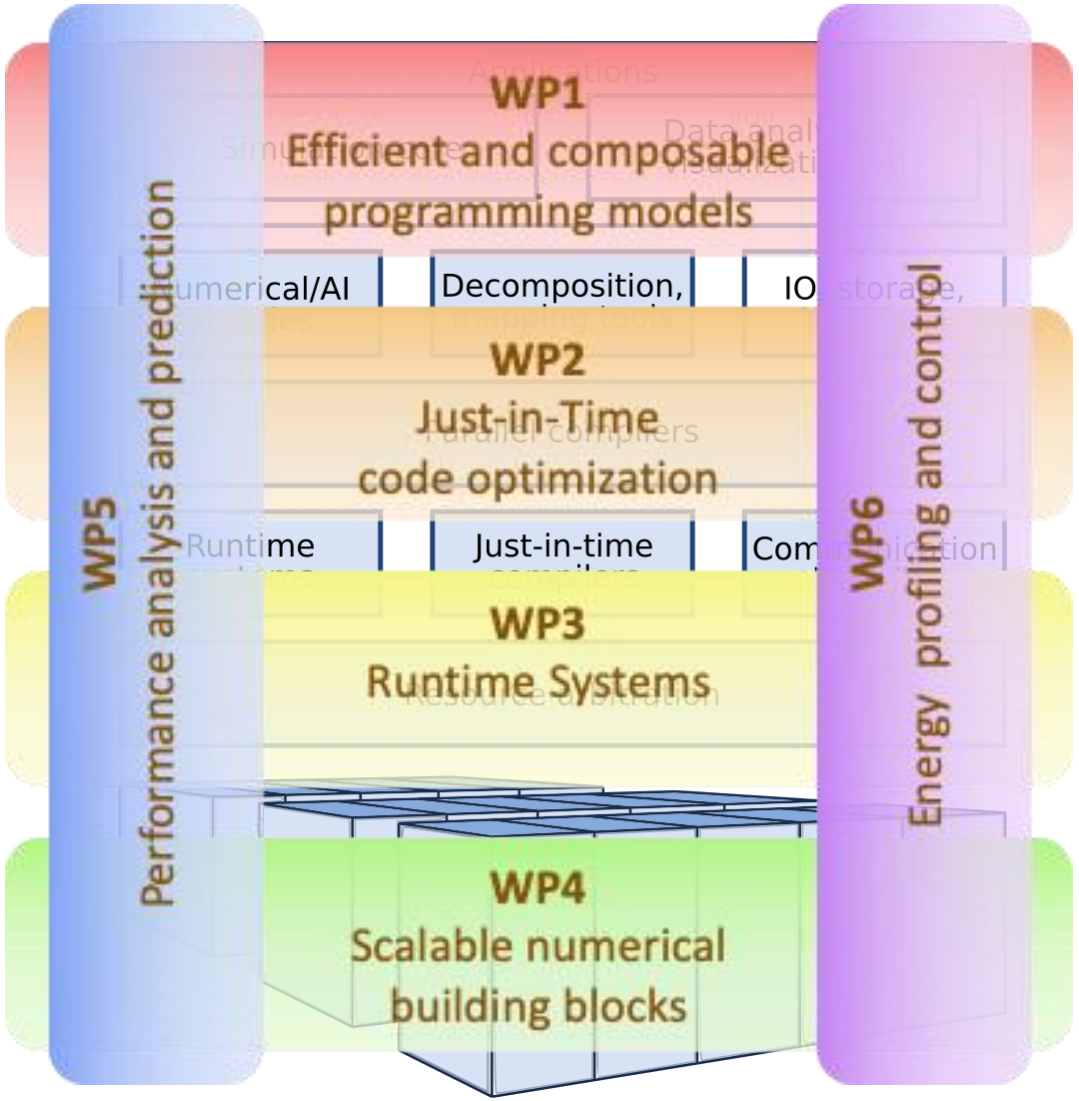
### Tasks:

1. *C++ complexity disambiguation for advanced optimizing and parallelizing code transformations*
2. *Tools for parallel heterogeneous scientific application at scale*
3. *Foundation of an HPC Composition Model*
4. *High level data description and partitioning for reusable parallel building blocks*

### Team:

*Inria, CEA, IFPEN*

The NumPEx software stack



# NumPEx / ExaSoft

*Exascale computing*

## WP1: Efficient and composable programming models

### Objectives:

- *Abstract the application (code and data) from the machines*
  - *Ease the porting to new machines*
  - *Ease the exploitation of machine heterogeneity*
  - *Ease debugging of complex software stack*
- *Improve code composability*
  - *Simplify code reuse to reduce the development*
  - *Improve the separation of concerns between domain experts and HPC experts*

### Tasks:

1. *C++ complexity disambiguation for advanced optimizing and parallelizing code transformations*
2. *Tools for parallel heterogeneous scientific application at scale*
3. *Foundation of an HPC Composition Model*
4. *High level data description and partitioning for reusable parallel building blocks*

### Team:

*Inria, CEA, IFPEN*

## Task 1.1

- *Exascale Architectures are heterogeneous architectures*

➔ *Extension to C++ language to deal with heterogeneous architectures*

- *Automatic optimizations of programs are hard to achieve in C++, due to language complexity*
- *Advanced control and data structures implemented in C++ make the work of the compiler extremely difficult*

➔ *Compiler extensions to improve C++ performances*

- *Use case and ecosystem:*
  - *Arcane Mini-apps are our main targets*
  - *Link with CEA initiative CExA on Kokkos*

## WP1: Efficient and composable programming models

### Objectives:

- *Abstract the application (code and data) from the machines*
  - *Ease the porting to new machines*
  - *Ease the exploitation of machine heterogeneity*
  - *Ease debugging of complex software stack*
- *Improve code composability*
  - *Simplify code reuse to reduce the development*
  - *Improve the separation of concerns between domain experts and HPC experts*

### Tasks:

1. *C++ complexity disambiguation for advanced optimizing and parallelizing code transformations*
2. *Tools for parallel heterogeneous scientific application at scale*
3. *Foundation of an HPC Composition Model*
4. *High level data description and partitioning for reusable parallel building blocks*

### Team:

*Inria, CEA, IFPEN (partner)*

### Task 1.2

- *Designing and providing a debugging tool to ease the conception and validation of scientific applications*
- *First target of this tool will be the C++ extension developed in Task 1.1.*
- *The tool will be based on SciHook*

### Task 1.3

- *Providing the foundations of an HPC composition framework*
- *Challenge is to provide an efficient composability model across a wide area of parallelism paradigms and hardware (GPGPU and task programming, networking and collective communication, etc)*

### Task 1.4

- *Contribute to the definition of high level data description models and associated partitioning mechanism with a focus on an efficient integration with the composition model*

# Questions ?