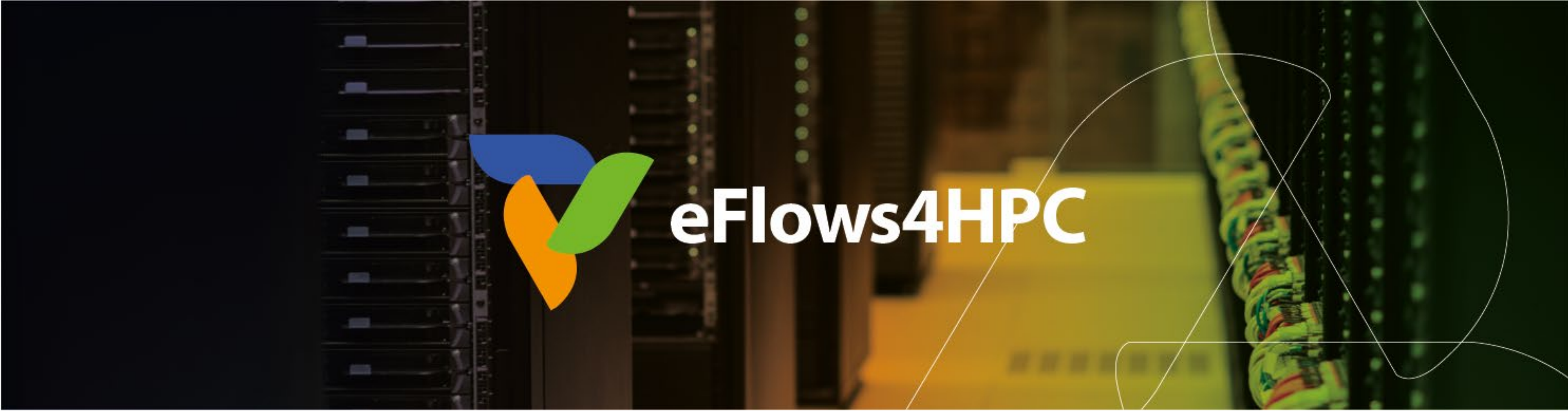


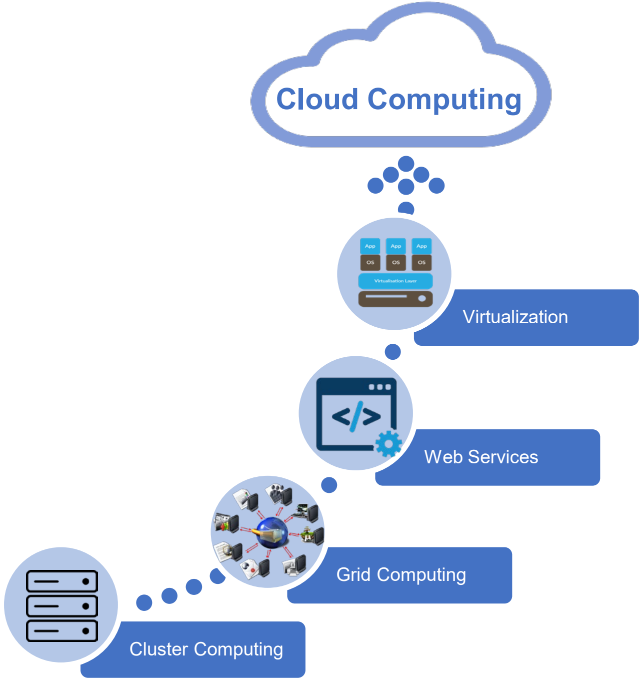
Introducing the FaaS model in Complex HPC Workflows: The eFlows4HPC approach

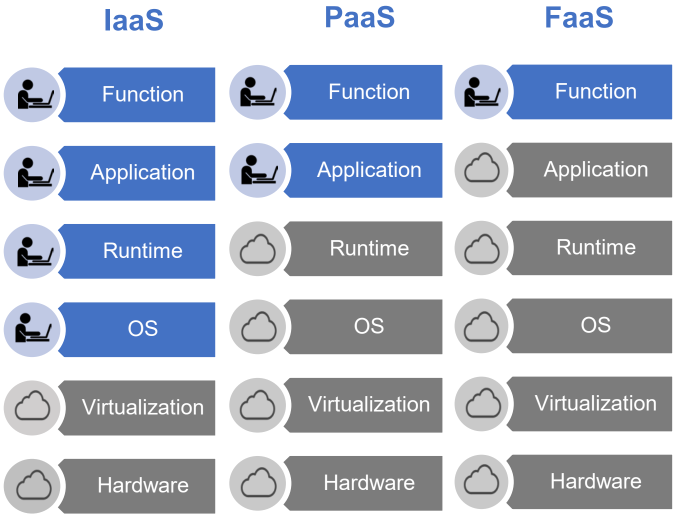
PDP Conference 2023

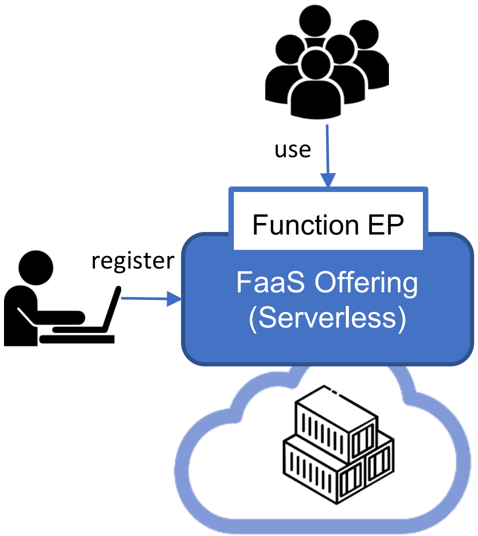
Jorge Ejarque (BSC)

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.

Distributed Computing Infrastructure Evolution

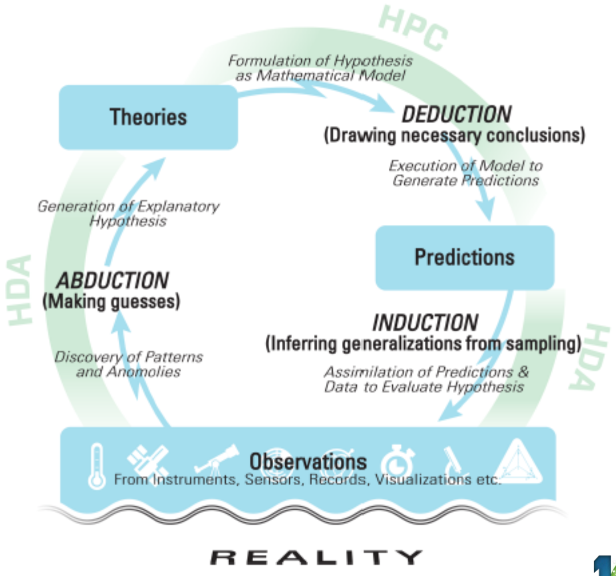






Evolution of the e-science workflows

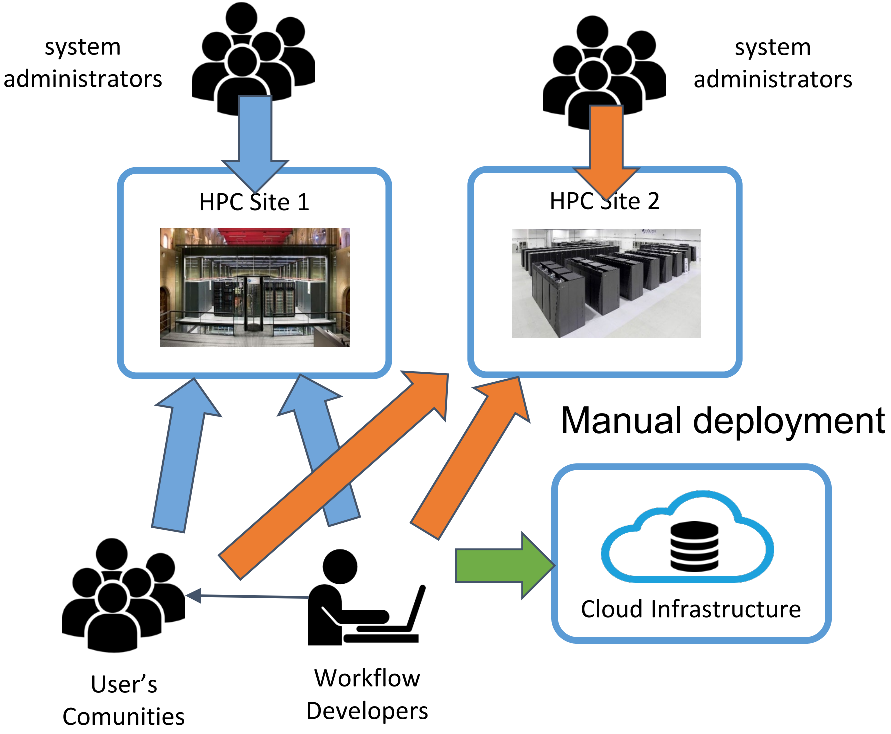
- **Combine Multiple frameworks**
  - Traditional HPC simulations
  - HPDA
  - Machine Learning
- **Integrate with data sources**
- **Dynamicity**
  - Workflow structure dynamically change according to preliminary results/events



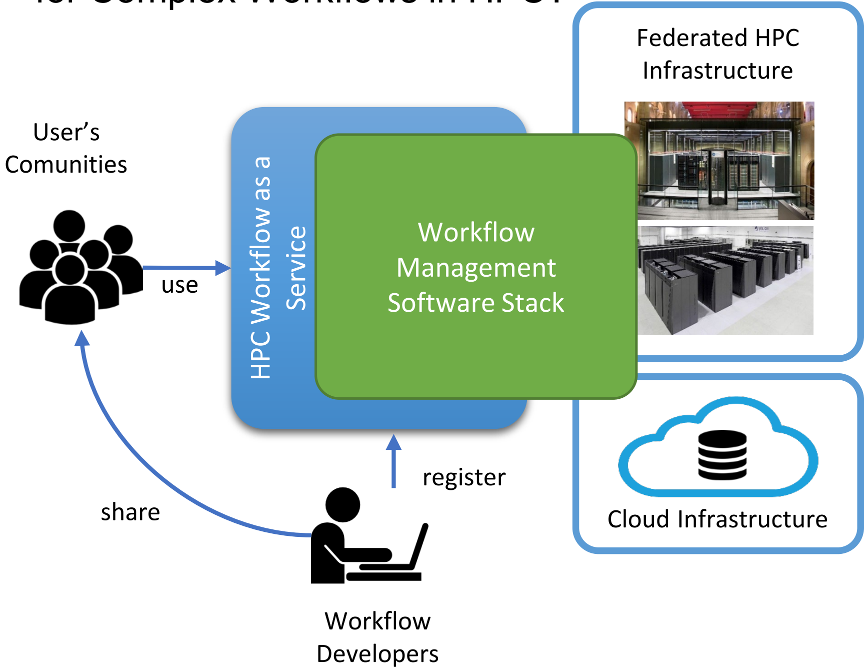
From "Big data and extreme-scale computing: Pathways to convergence-toward a shaping strategy for a future software and data ecosystem for scientific inquiry"

HPC Environments

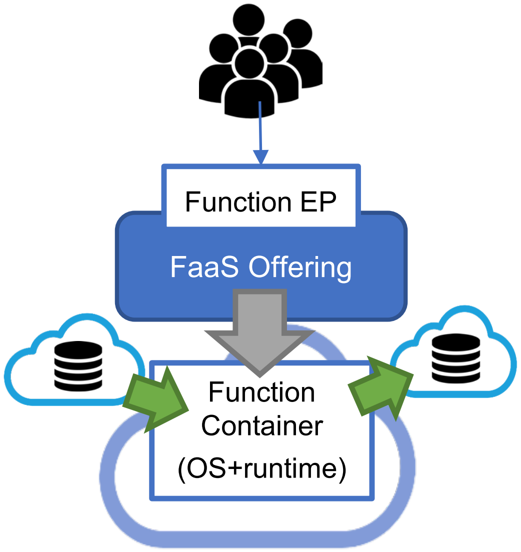
Current approach



Can we apply something like FaaS for Complex Workflows in HPC?



FaaS vs. HPCWaaS

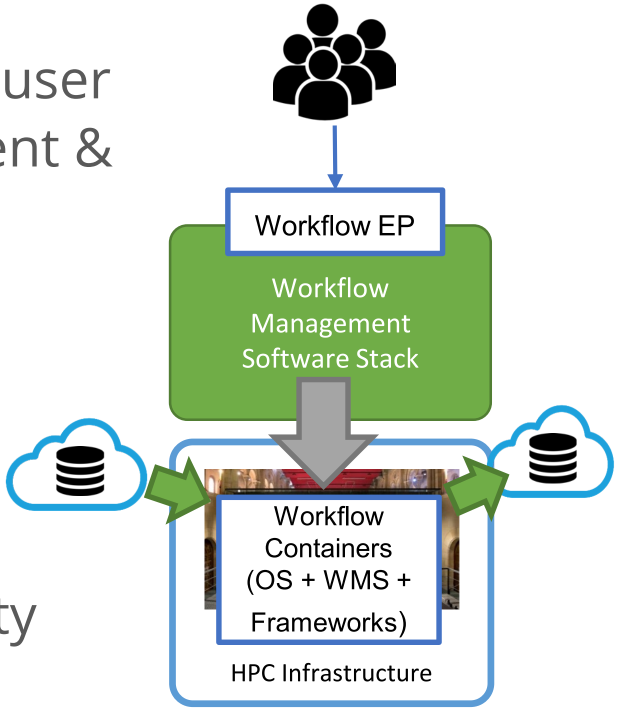


Similarities

- Easy to use for final user
- Automate deployment & execution
- Data integration
- Containers

Differences

- Restrictions
- Deployment and Execution Complexity
- Performance

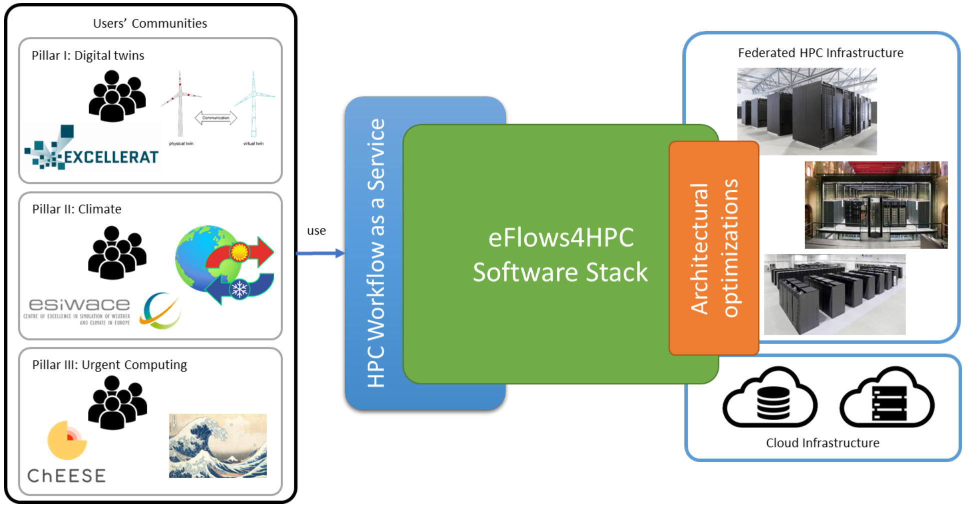


EuroHPC

- **EuroHPC aims at developing a World Class Supercomputing Ecosystem in Europe**
    - Procuring and deploying exascale, pre-exascale and petascale systems in Europe
- **Developing Software Environments**
    - Running large and complex applications in these systems
- **eFlows4HPC funded under call EuroHPC-02-2019:**
    - High Performance Computing (HPC) and data driven HPC software environments and application oriented platforms

eflows4HPC

- **Software tools stack that make it easier the management of complex workflows:**
  - Combine different frameworks
    - HPC, AI + data analytics
  - Reactive and dynamic workflows
    - Automatic workflow steering
  - Full lifecycle management
    - Not just execution
    - Data logistics and Deployment
- **HPC Workflows as a Service:**
  - Mechanisms to make it easier the use and reuse of HPC by wider communities



- **Architectural Optimizations:**
  - Selected HPC – AI Kernels Optimized for GPUs, FPGA, EPI
- **Validation Pillar's**
  - Workflows of users representing CoEs

Pillar I: Manufacturing

CIMNE

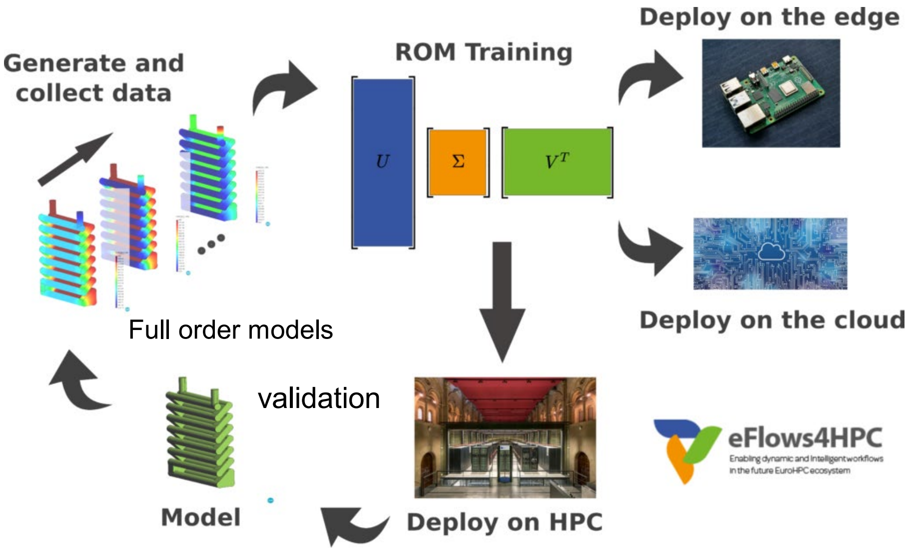
Inria

SISSA
Scuola Internazionale Superiore di Studi Avanzati

SIEMENS

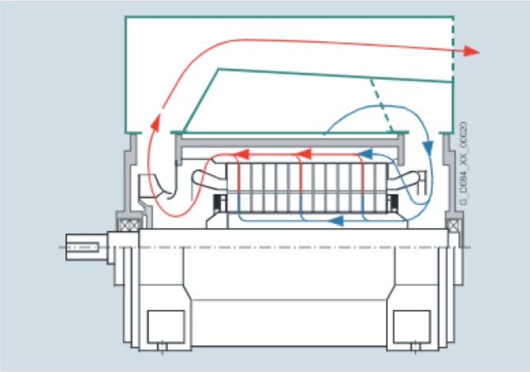
KRATOS Multi-Physics

Upgrade your meshes

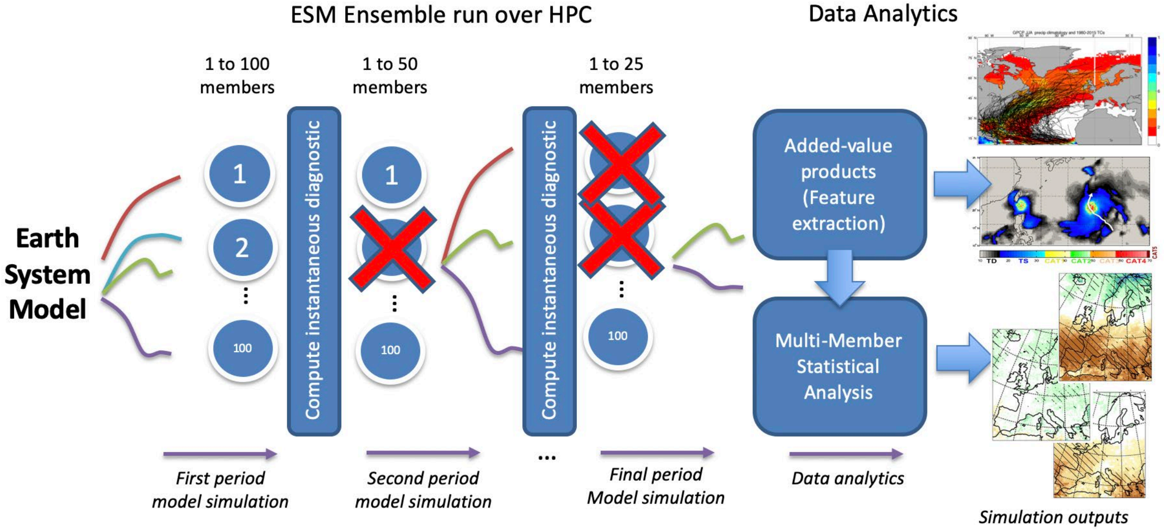


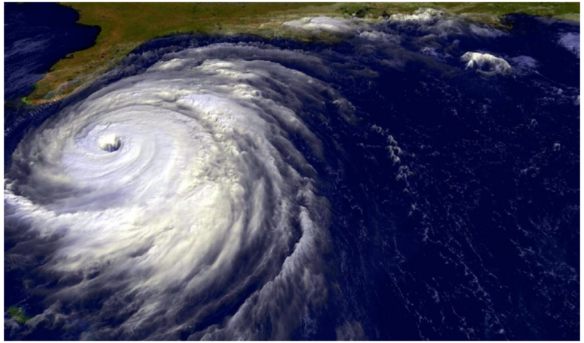
Pillar I focuses on the construction of DigitalTwins for the prototyping of complex manufactured objects:

- Integrating state-of-the-art adaptive solvers with machine learning and data-mining
- Contributing to the Industry 4.0 vision



Pillar II: Climate



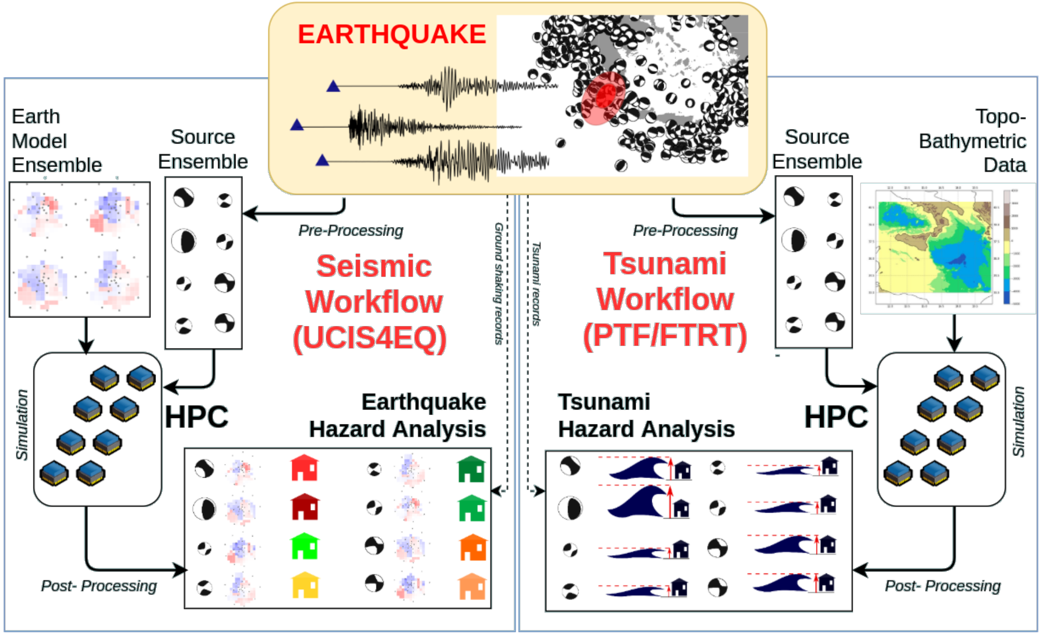


Dynamic (AI-assisted) workflow

HPDA & ML/DL

- Perform climate predictions: temperature, precipitation or wind speed
- AI-assisted pruning of the ESM workflow
- Study of Tropical Cyclones (TC) in the North Pacific, with in-situ analytics

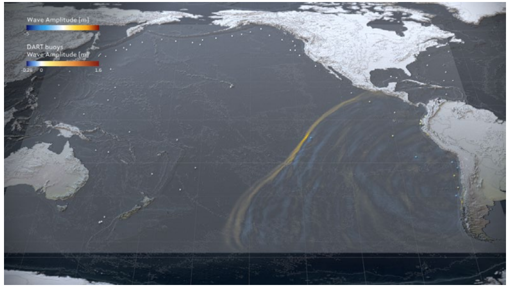
Pillar III: Urgent computing for natural hazards



Pillar III explores the modelling of natural catastrophes:

- Earthquakes and their associated tsunamis shortly after such an event is recorded
- Use of AI to estimate intensity maps
- Use of DA and AI tools to enhance event diagnostics
- Areas: Mediterranean basin, Mexico, Iceland and Chile

Tsunami-HySEA GPU-based code



Barcelona Supercomputing Center – Centro Nacional de Supercomputación

ETH zürich

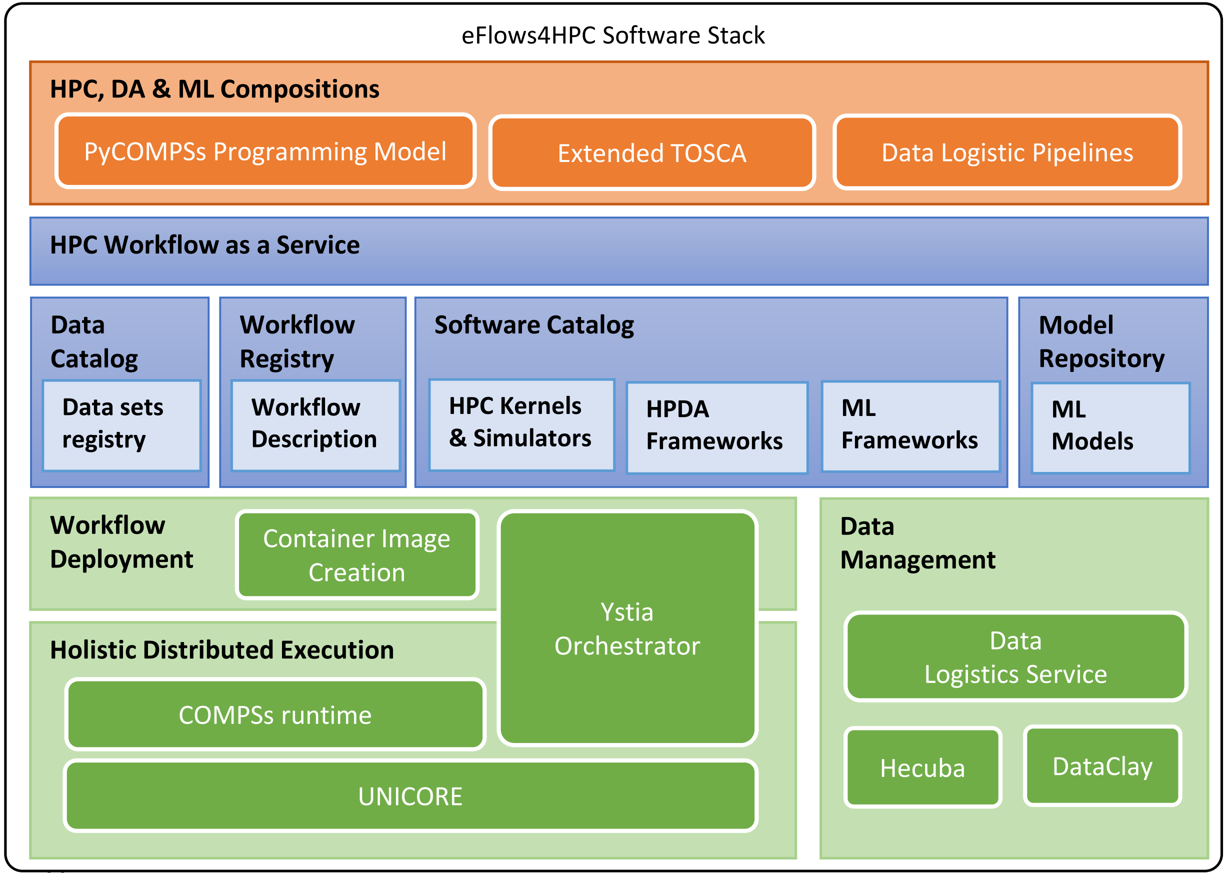
Istituto Nazionale di Geofisica e Vulcanologia

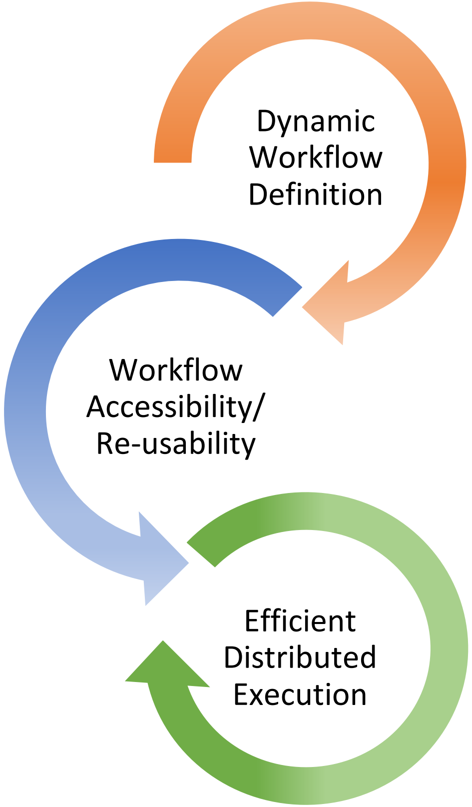
PSNC

Universidad de Málaga

NGI

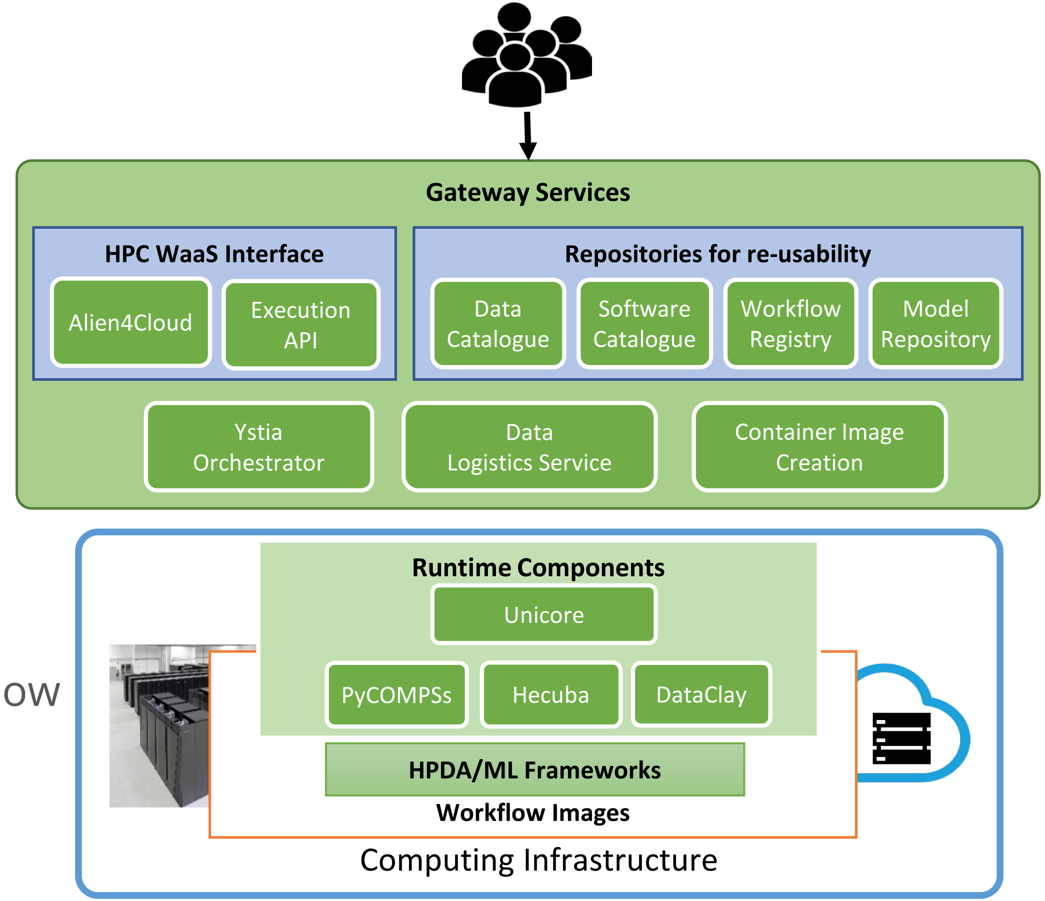
Software Stack overview



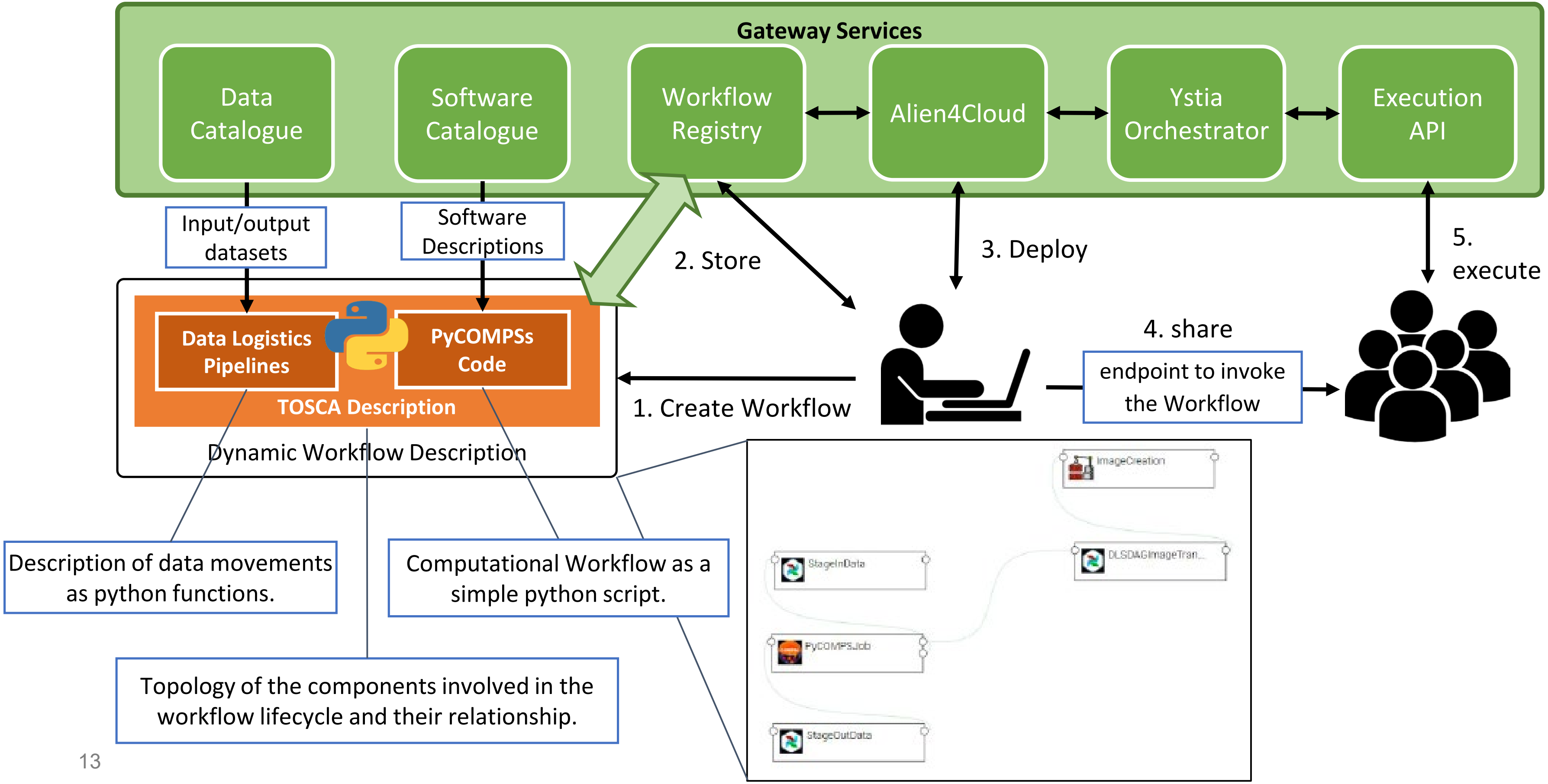


eFlows4HPC software stack and HPCWaaS

- **Gateway services**
  - Components deployed outside the computing infrastructure.
  - Managing external interactions and workflow lifecycle
- **Runtime Components**
  - Deployed inside the computing infrastructure to manage the workflow execution



Development Overview



PyCOMPSs

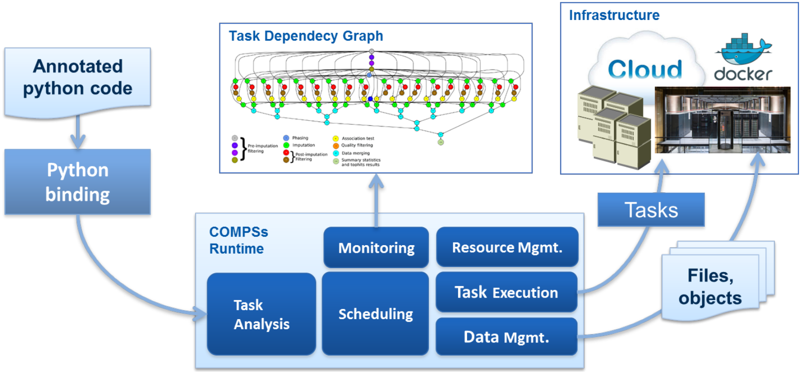
- **Sequential programming, parallel execution**
- **Python Code + annotations/hints**
  - To identify tasks and directionality of data
- **Builds a task graph at runtime that express potential concurrency**
- **Offers to applications the illusion of a shared memory in a distributed system**
- **Agnostic of computing platform**

```python
@task()
def word_count(block):
    ...
    return res

@task(f_res=INOUT)
def merge_count(f_res, p_res):
    ...
```

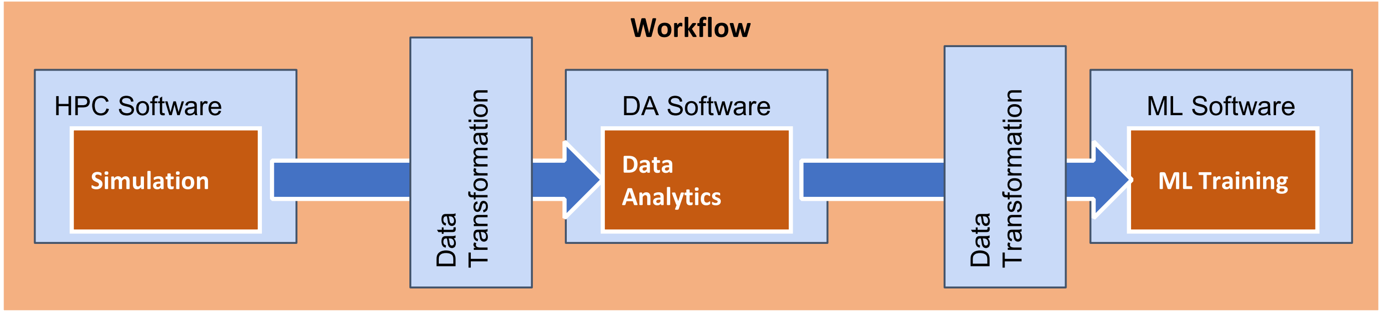
(a) Task annotation example

```python
for block in data:
    p_result = word_count(block)
    reduce_count(result, p_result)
result = compss_wait_on(result)
```

(b) Main code example



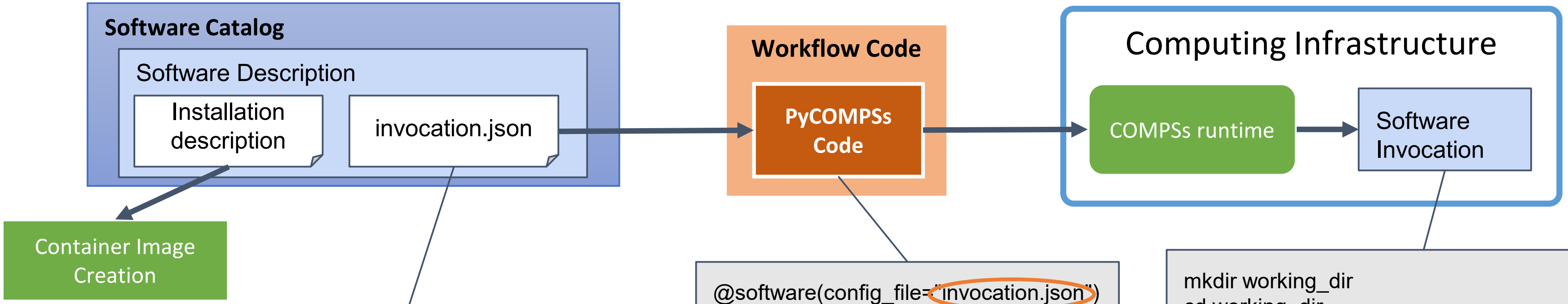
Interfaces to integrate HPC/DA/ML



- **Goal:**
  - Reduce glue code
  - Focus on the functionality, not in the integration
  - Reusability
- **First phase:** software integration
- **Second phase:** data transformations

```
@data_tranformation(input_data, function)
@software(invocation description)
def data_analytics (input_data, result):
   pass

#Worflow

simulation(input_cfg, sim_out)
data_analytics(sim_out, analysis_result)
ml_training(analysis_result, ml_model)
```

Software Invocation description




```
{
  "type":"mpi",
  "properties":{
     "runner": "mpirun",
     "processes": "$SW_PROCS"
     "binary": "mpi_sofware.x",
     "params": "-d {{param}}",
     "working_dir": "{{working_dir}}"},
  "prolog":{
       "binary":"mkdir",
       "params":"{{working_dir}}"},
    "epilog":{
       "binary":"tar",
       "params":"zcvf {{out_tgz}}" {{working_dir}}},
  "constraints":{
     "computing_units": $SW_THREADS}
}
```

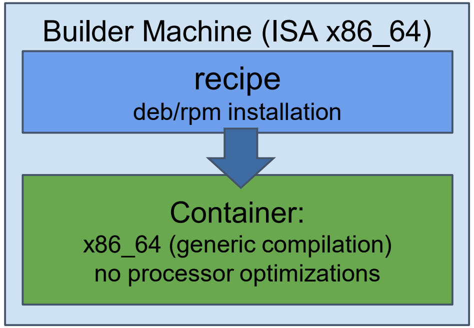
```
@software(config_file="invocation.json")
def mpi_exec(work_dir, param, out_tgz):
    pass

#Call to the task function:

mpi_exec('my_folder', 'hello_world')
```

```
mkdir working_dir
cd working_dir
export OMP_NUM_THREADS=$SW_THREADS
mpirun -n $SW_PROCS mpi_software.x -d param
tar zcvf out_tgz working_dir
```

- **Converts a python function to software invocation as a PyCOMPSs task**
- **Reused in different workflows**

Containers and HPC

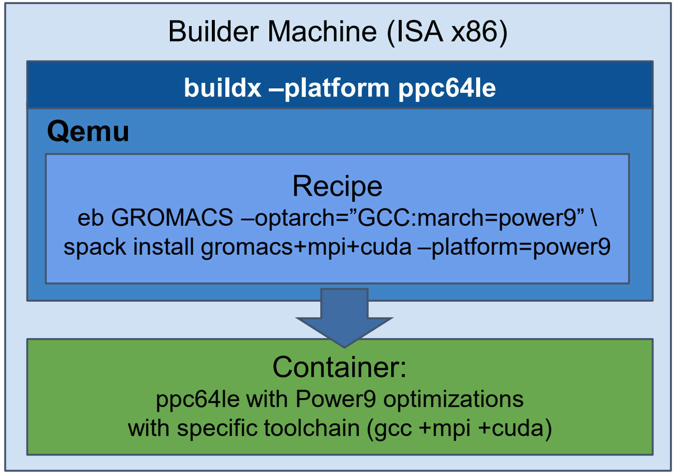
Standard container image creation



- **Simplicity for deployment**
  - Just pull or download the image
- **Trade-Off performance/portability**
  - Architecture Optimizations
- **Accessing Hardware from Containers**
  - MPI Fabric /GPUs
- **Host-Container Version Compatibility**

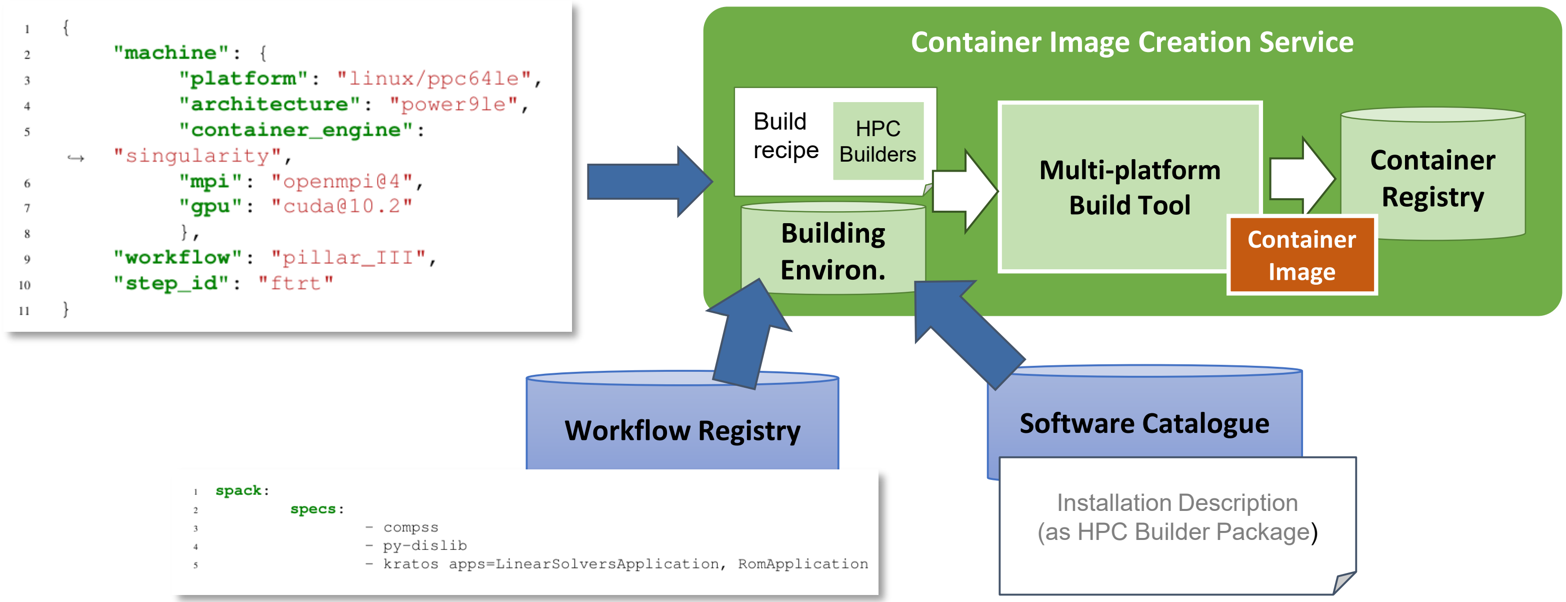
HPC Ready Containers

eFlows4HPC approach



- **Methodology to allow the creation containers for specific HPC system**
  - Leverage HPC and Multi-platform container builders
- **It is tight to do by hand but let's automate!**

HPC Ready Containers

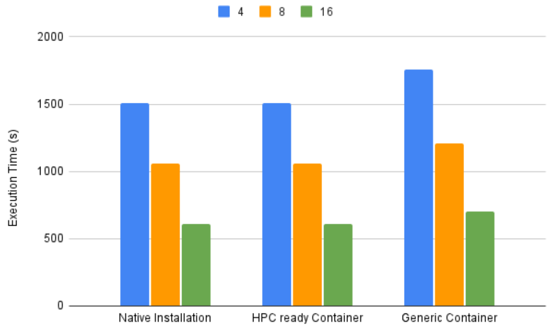


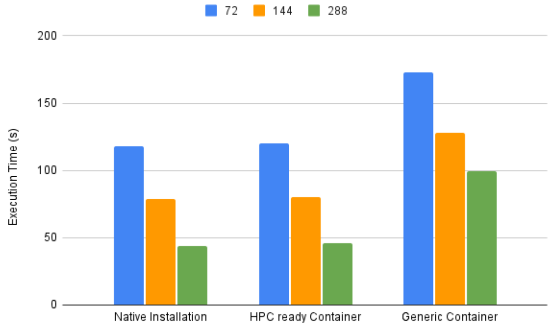
HPC-Ready Containers

Kratos Multiphysics (shared memory)

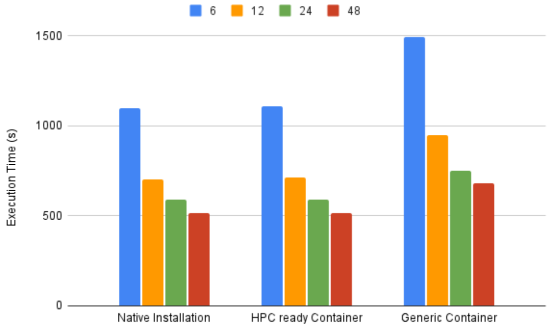
FESOM2 (MPI)

Nord3

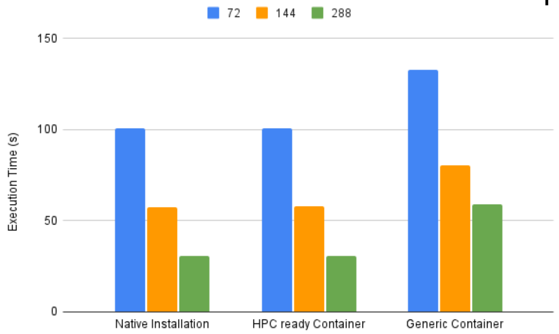




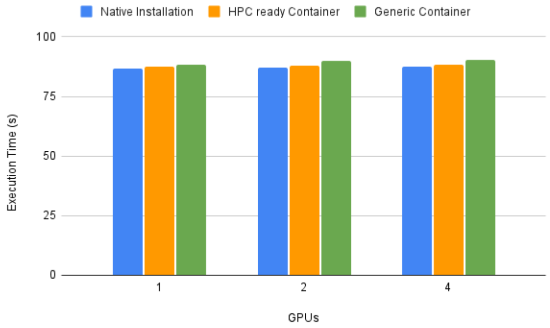
MN4



CTE-Power



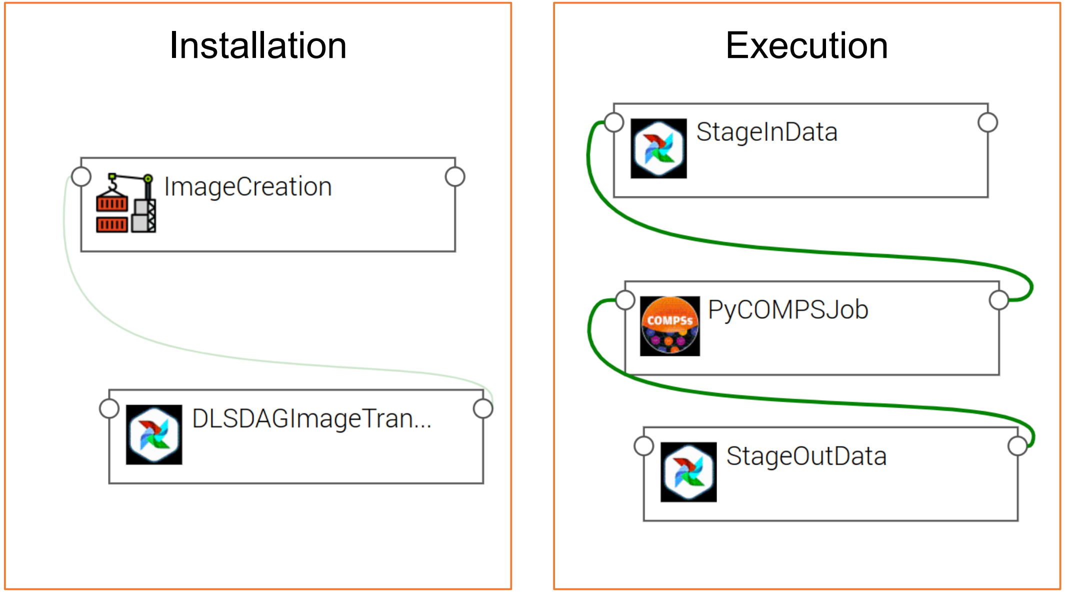
Tsunami-HySEA



TOSCA Modelization

eFlows4HPC

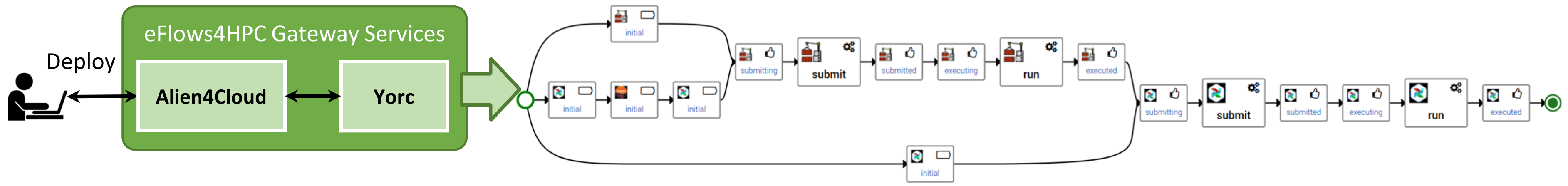
Topology of the different components involved in the Workflow lifecycle



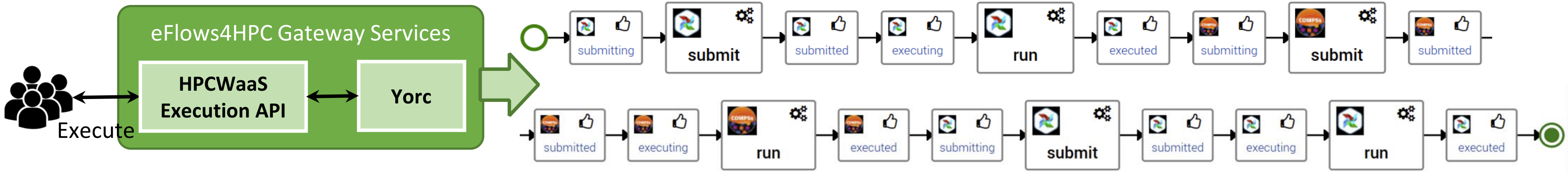
TOSCA Modelization

eFlows4HPC

Application deployment workflow (done once)



End-User workflow (multiple executions)



Conclusion

- **eFlows4HPC**
  - Software stack and HPCWaaS
    - manage complex workflows in the whole lifecycle
    - Enable reusability of workflows and their components
    - Facilitate the deployment through HPC-Ready containers
    - Facilitates the accessibility of HPC systems
    - Reduce workflow management efforts

Thank you

eFlows4HPC

Enabling dynamic and Intelligent workflows in the future EuroHPC ecosystem

www.eFlows4HPC.eu

@eFlows4HPC   eFlows4HPC Project

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.