An introduction to verification, validation and uncertainty quantification for computational applications

Derek Groen, Diana Suleimenova, Laura Harbach, David Coster, Wouter Edeling, Lourens Veen, Bartosz Bosak, Serge Guillas, Jon McCullough, Alireza Jahani, Maziar Ghorbani and Peter Coveney

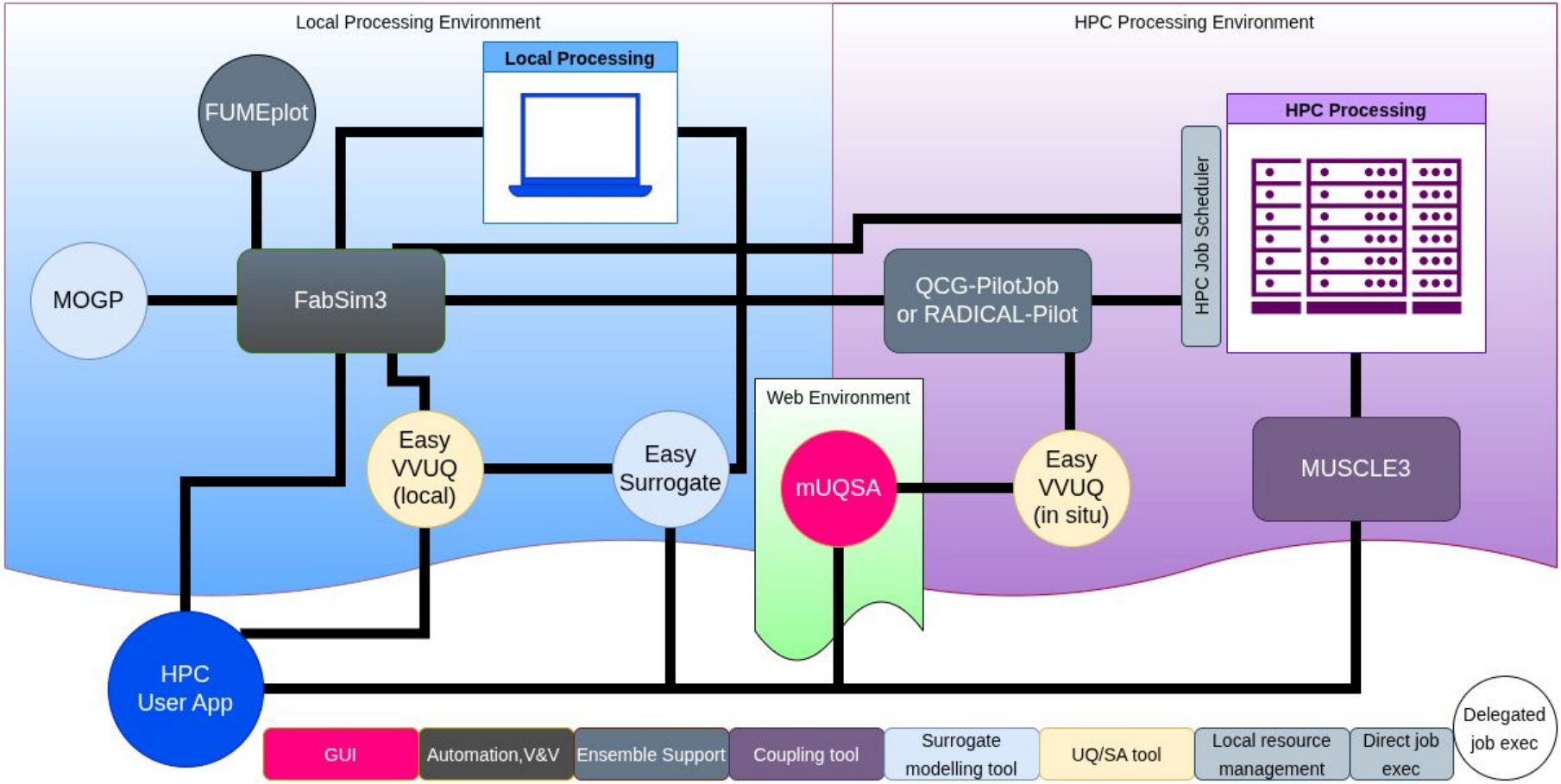
SEAVEA toolkit introduction

https://www.seaveatk.org

SEAVEA Project (2021–now)

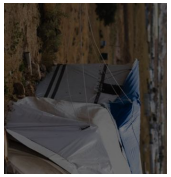
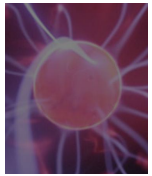
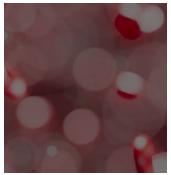
Key topics:

- Uncertainty Quantification - Measure variability of results
- Sensitivity Analysis - Measure variability of results due to specific assumptions.
- Validation - Reality is consistent with the computer model.
- Ensemble Forecasting - Variability of results with scenario-steered uncertainties.
- Surrogate Modelling - Faster modelling with reduced order models.
- Verification - Computer model is consistent with the conceptual model.
- UQ for AI - Understand variability of results in AI approaches.
- Uncertainty *Qualification* - Understand non-quantitative weaknesses in models.


Local Processing Environment
HPC Processing Environment
Local Processing
HPC Processing
FUMEplot
MOGP
FabSim3
QCG-PilotJob
or RADICAL-Pilot
HPC Job Scheduler
Web Environment
mUQSA
Easy
VVUQ
(local)
Easy
Surrogate
Easy
VVUQ
(in situ)
MUSCLE3
HPC
User App
GUI
Automation,V&V
Ensemble Support
Coupling tool
Surrogate
modelling tool
UQ/SA tool
Local resource
management
Direct job
exec
Delegated
job exec

SEAVEA Application Domains

- Materials
- Turbulence
- Biomedicine
- Fusion
- Environmental Sciences
- Epidemiology
- Human Migration
- Infrastructure Modelling

SEAVEA

Brunel University of London

UCL

THE STATE UNIVERSITY OF NEW JERSEY RUTGERS

CWI Centrum Wiskunde & Informatica

OAK RIDGE National Laboratory

IPP

UK Atomic Energy Authority

Argonne NATIONAL LABORATORY

RIKEN

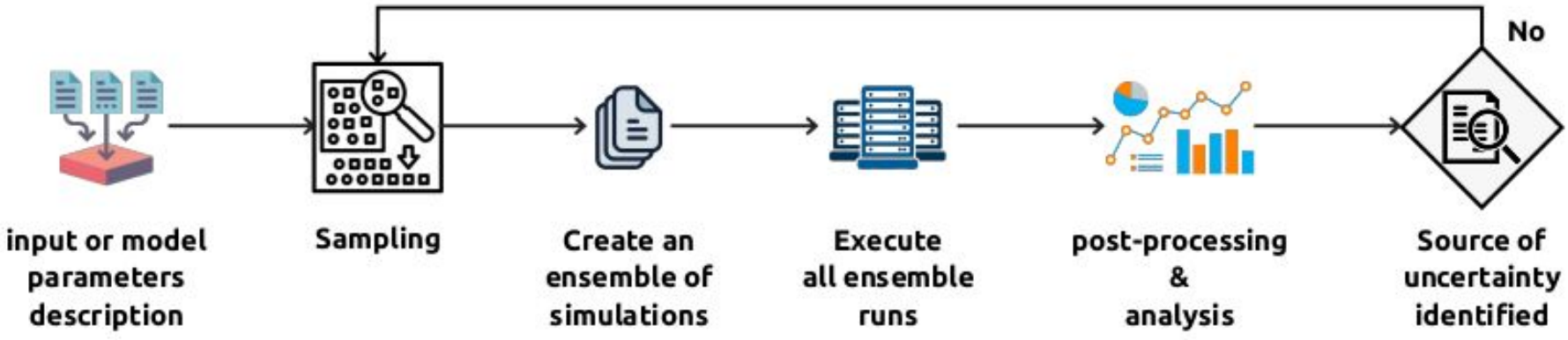
Imperial College London

Save the Children

PSNC

EasyVVUQ: https://easyvvuq.readthedocs.io

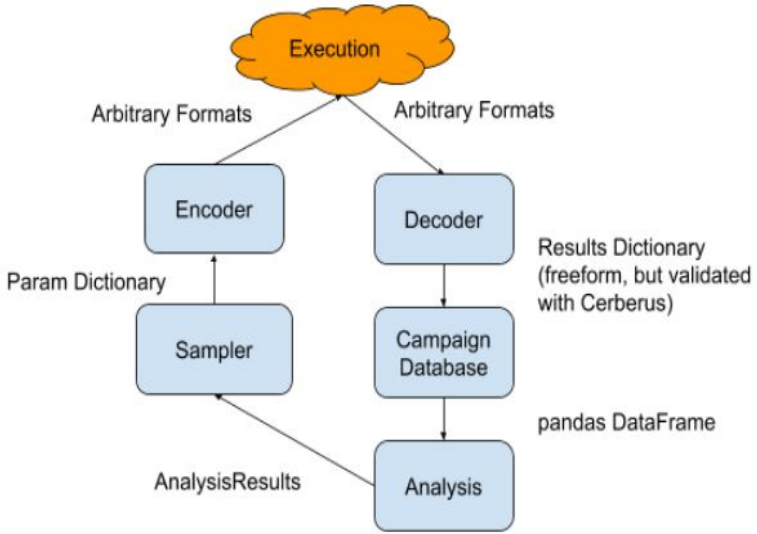
- Aims to make it as easy as possible to implement advanced techniques for uncertainty quantification for existing application codes (or workflows).
- Primary focus on non-intrusive UQ techniques, where many model instances are run to quantify uncertainties and analyse parameter sensitivities.



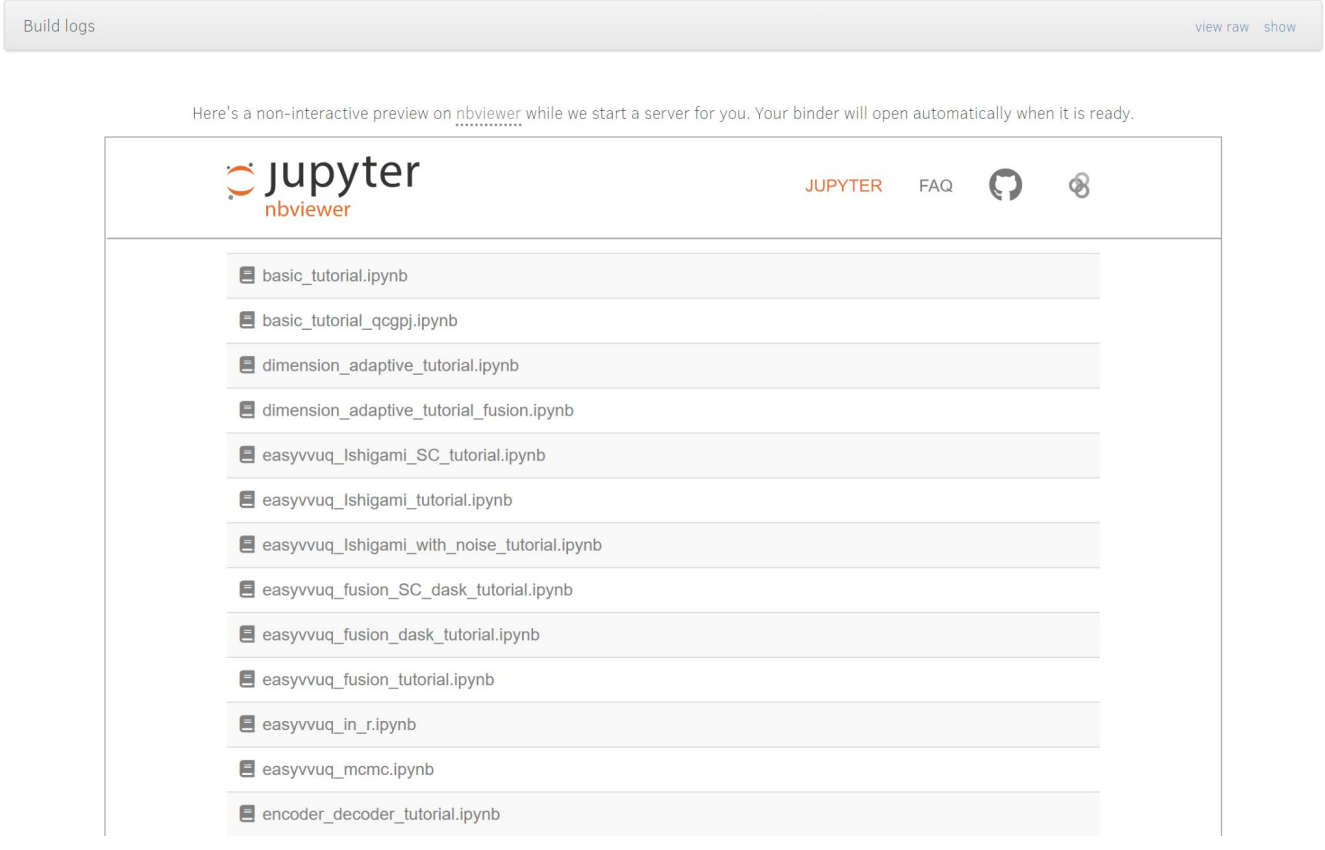
Basic UQ workflow

EasyVVUQ: Enhances Workflow

- Integrated tightly with the rest of Python scientific computing infrastructure.
  - E.g. access through a wide range of sampling approaches using ChaosPy.
- Adapts to users existing code via extension of encoder and decoder classes (e.g. Fusion and Urban Air).
- Minimal boiler-plate code required (minimal set-up overhead).
- Takes care of bookkeeping (creating folders for storing results for example), collects other meta-info in a database.



https://mybinder.org/v2/gh/UCL-CCS/EasyVVUQ/dev?filepath=tutorials

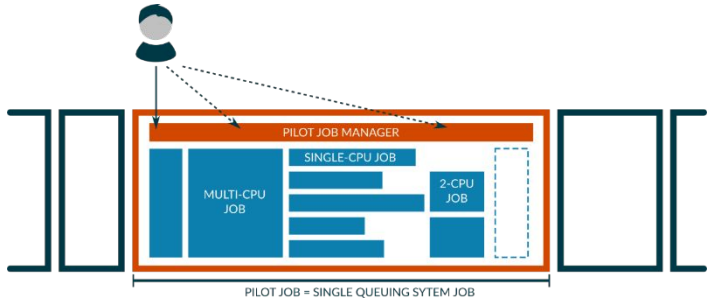


Execution of UQ on HPC machines

UQ procedures require numerous HPC jobs

HPC systems are configured to prefer large and long jobs over many small and short jobs

How to support UQ scenarios?



QCG PILOT JOB

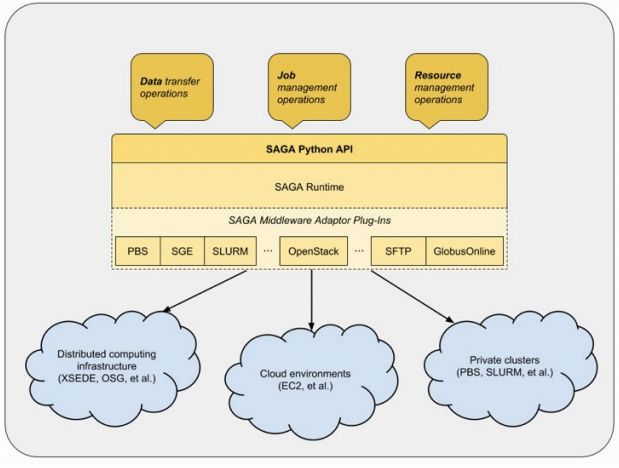
R

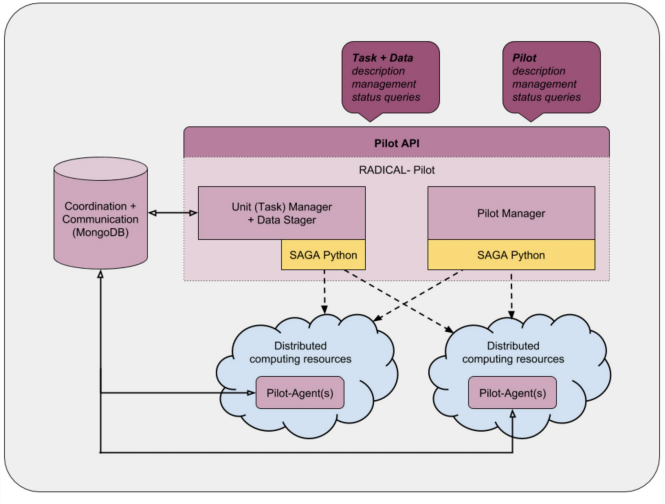
QCG-PilotJob: https://qcg-pilotjob.readthedocs.io

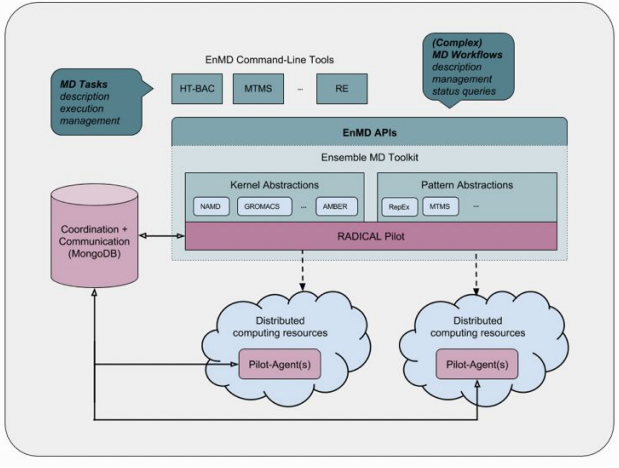
- Dynamic scheduling and running of tasks inside a single queuing system allocation (e.g., SLURM).
- Support for task dependencies and iterations.
- Different modes of usage:
  - static (description in JSON file),
  - dynamic (Python API and remote API).
- Resuming mechanism (can be used after task failure or premature end of allocation).
- Local mode of execution that enable usage of the system not only on HPC systems.
- Good scalability thanks to partitioning mechanism.

RADICAL Cybertools: https://radical-cybertools.github.io

- Promoting a standards-based, abstraction-driven approach to High-Performance Distributed Computing.
- Architected for scalable, interoperable and sustainable approaches to support science on a range of high-performance and distributed computing systems.







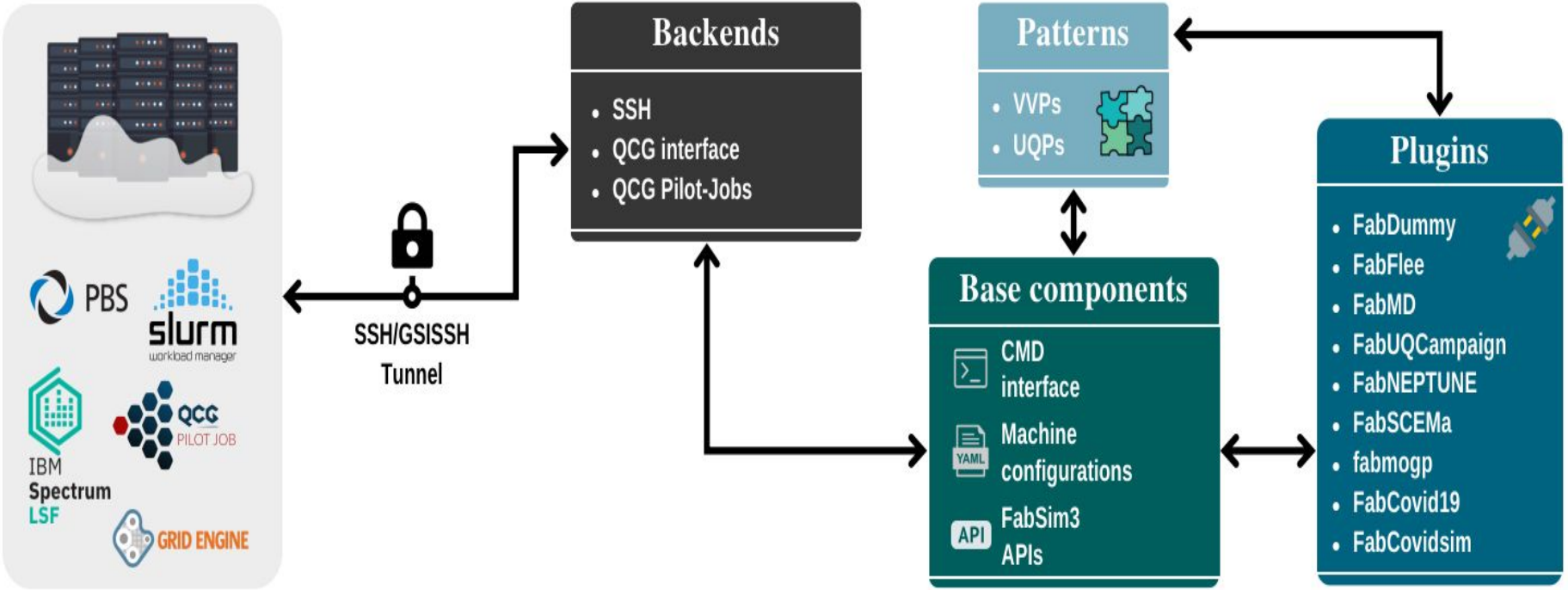
FabSim3: https://fabsim3.readthedocs.io

1. Reduce manual effort by providing flexible automation.
2. Reduce human error by introducing consistency across different activities.
3. Automatically curate a comprehensive environment for each simulation run.
4. Curate essential machine- and application-specific knowledge across the community.
5. And, as a result of 1-4: enable efficient remote access to HPC and other resources using a user-side local tool.

```
476   archer2:
477     <<: *Eagle_Default_Config
478     <<: *QCG_PilotJob
479     remote: "archer2" #ARCHER2 always requires multiplexing, so here we need an alias.
480     budget: "d137"
481     project: "d137"
482     corespernode: 128
483     run_command: "srun -n $cores"
484     qos_name: "standard"
485     # list of available partitions : sinfo --Format=partition
486     job_dispatch: "cd /work/$project/$project/$username ; sbatch"
487     partition_name: "standard"
488     batch_header: slurm-archer2
489     PJ_header: archer2-PJ-header
490     batch_header_install_app : archer-app
491     # all required files at runtime, must be on the /work filesystem,
492     # in ARCHER2, there is not access to /home on compute side
493     home_path_template: "/work/$project/$project/$username"
```

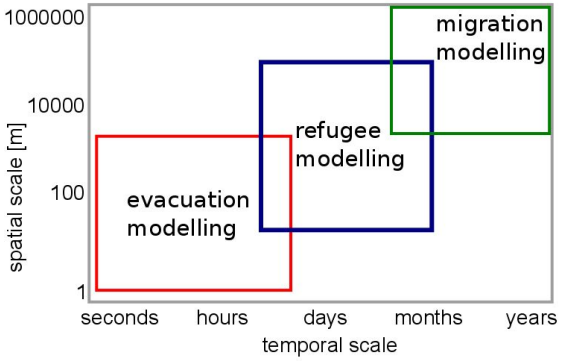
machines.yml

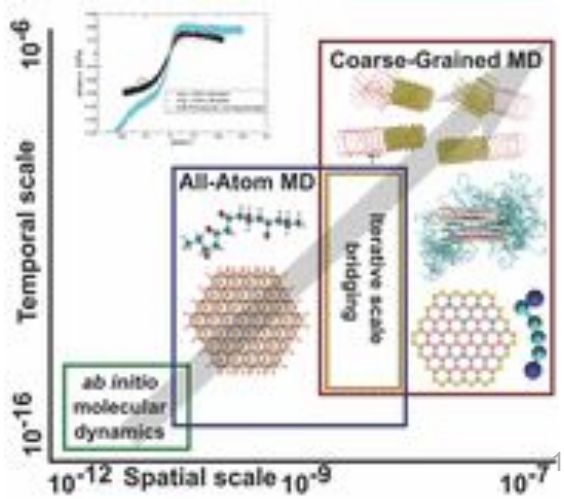
FabSim3 architecture



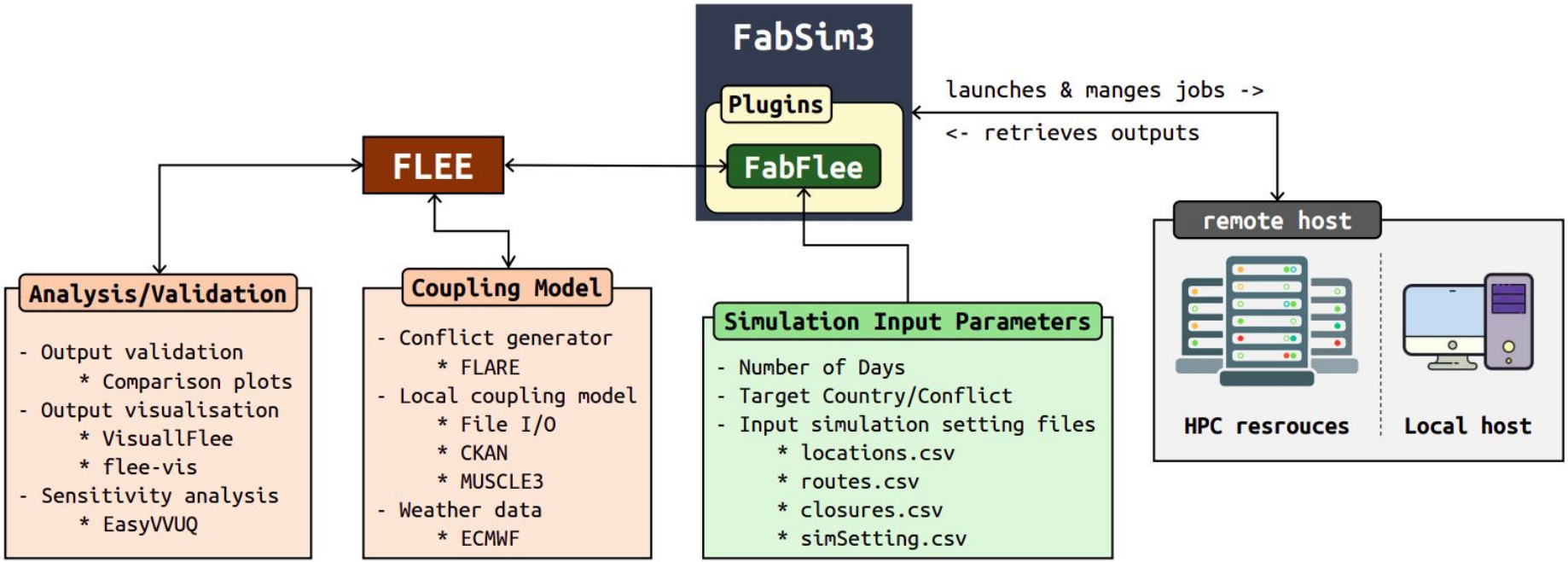
FabSim3 plugins

- FabDummy
  - Example plugin to help reduce learning curve.
- FabFlee
  - Automates refugee simulations. In use for research purposes.
- FabMD
  - Ensembles and UQ for materials.
- FabUQCampaign
  - Plugin for performing UQ using stochastic collocation.
- FabCovid19, FabCovidSim
  - Plugins for two epidemiology applications
- Other plugins:
  - https://github.com/djgroen/FabSim3/blob/master/fabsim/deploy/plugins.yml



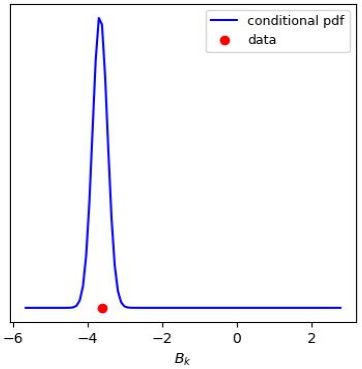


Plugin Example: FabFlee



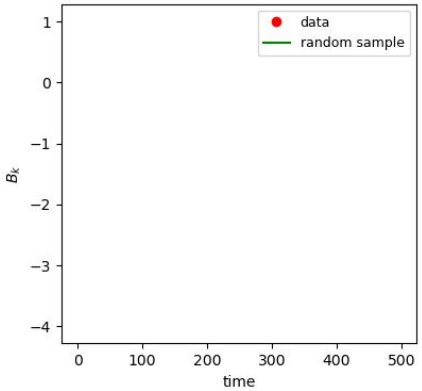
EasySurrogate: https://github.com/wedeling/EasySurrogate

- ❏ Toolkit used for creating surrogate models.
- ❏ Aims to provide different surrogate methods used to replace the micro model in a multiscale simulation.
- ❏ Follows similar design structure as EasyVVUQ.



Example surrogate:

+ Top: a probability density surrogate of microscopic scales, conditional on macroscopic features.
+ Bottom: time evolution of training data and random samples from the surrogate.

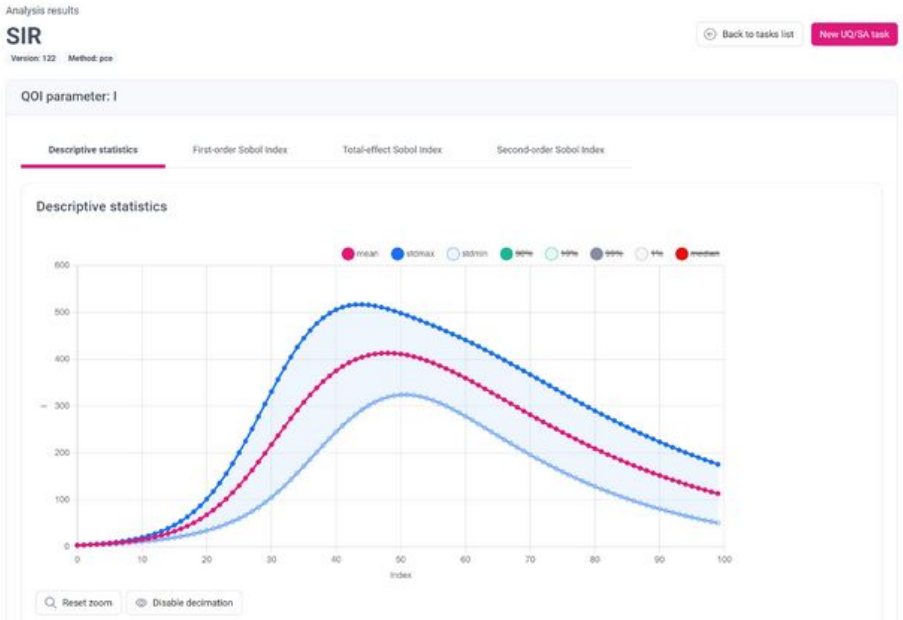


MUSCLE3: https://muscle3.readthedocs.io

- The third incarnation of the Multiscale Coupling Library and Environment.
- Aims to create easily coupled multiscale simulations, and to then enable efficient uncertainty quantification of such models using advanced semi-intrusive algorithms.
- Implementation is available in Python, C++ (with/without MPI), Fortran (with/without MPI).

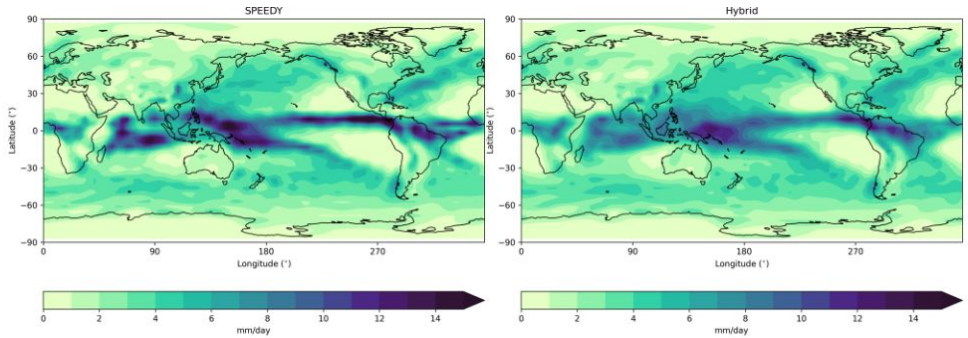
mUQSA: https://muqsa.multiscale.pionier-lab.pionier.net.pl/en/

- Provides a comprehensive system for the automated UQ and SA of computational models.
- Implements proven methods such as
  - Monte Carlo (MC), Polynomial Chaos Expansion (PCE), and Stochastic Collocation (SC).
- Offers an intuitive web interface and efficient backend layer.
- Supports development, execution, and analysis of the results.



Multi-Output Gaussian Process emulator (MOGP)

- Python package for fitting Gaussian Process Emulators to computer simulation results.
- Optimises hyperparameter values, and makes predictions on unseen data.
- Implements experimental design, dimension reduction, and calibration tools to enable modellers to understand complex computer simulations.
- https://mogp-emulator.readthedocs.io/

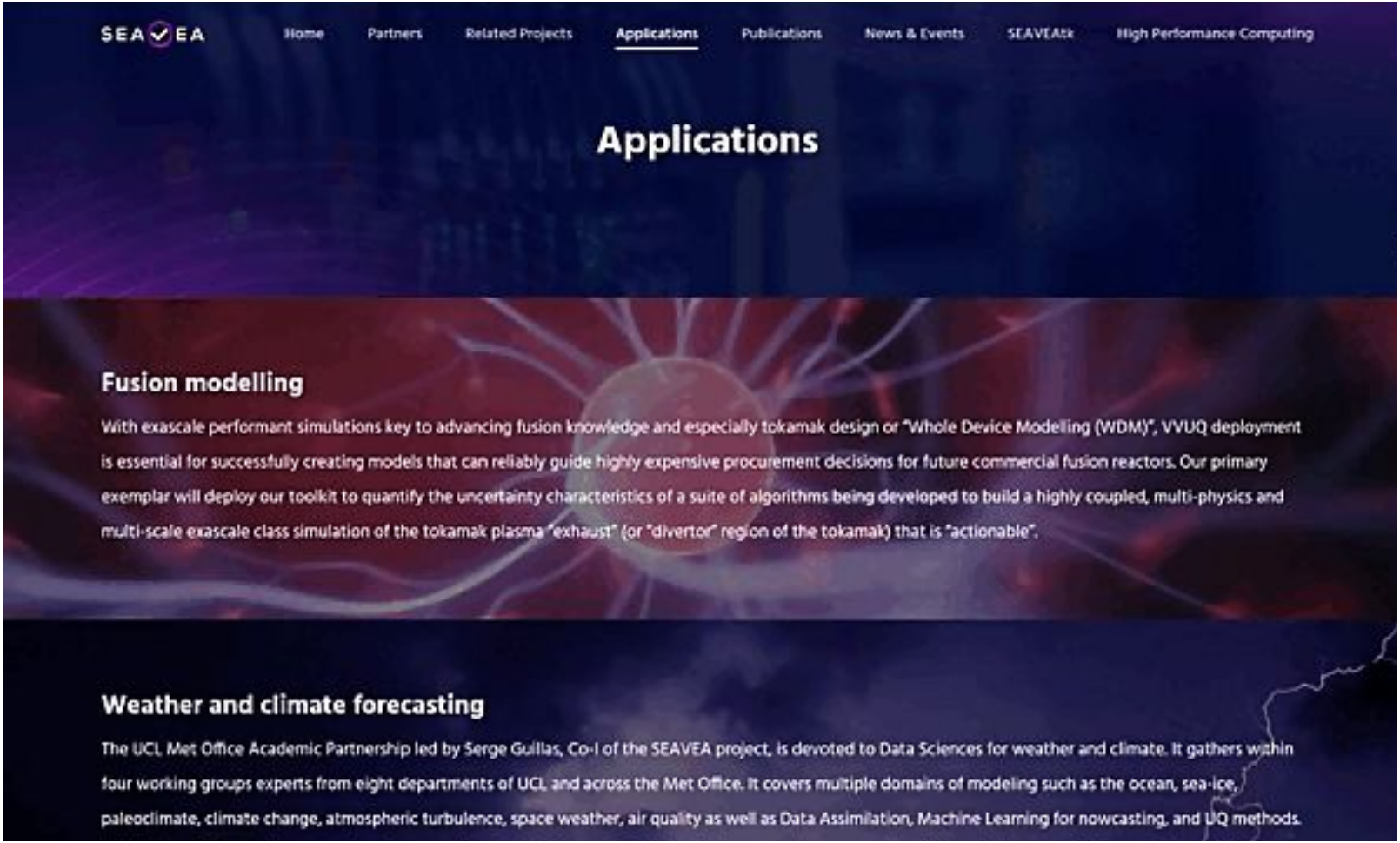


Giles, D., Briant, J., Morcrette, C.J. and Guillas, S., 2024. Embedding machine-learnt sub-grid variability improves climate model precipitation patterns. *Communications Earth & Environment*, *5*(1), p.712.

SEAVEA toolkit

- EasyVVUQ → UQ+SA,
- FabSim3 → VV + automation,
- EasySurrogate → Surrogate modelling for multiscale simulations,
- MUSCLE3 → Code coupling + UQ,
- mUQSA → Web-based GUI for Multipurpose UQ + SA,
- QCG-PilotJob → Run 1000s of jobs in a single allocation,
- RADICAL-Cybertools → Abstraction-driven approach to HPC.
- mogp_emulator → Surrogate modelling for fitting Gaussian Process Emulators,

The SEAVEA toolkit is open source and freely available to use with any application, using any programming language.



https://www.seaveatk.org/applications

Questions?

https://www.seaveatk.org