# EGI-ACE HPC integration: Fusion use case pilot

Shaun de Witt

UK Atomic Energy Authority

**Dissemination level:** Public

**Disclosing Party:** UKAEA

**Recipient Party:** EGI-ACE Project

# Overview

- Use case
  - Fusion Data Space Provider
  - JOREK
- Technology
  - PROMINENCE
- Requirement for HPC
- Experience integrating UK HPC

# Use case

*Fusion Data Space Provider*

- Building NN-based surrogate models in an efficient & effective manner
  - Traditional approach involves arbitrary scans across a range of parameters
  - May be unaware of more complex behaviour & nuances
- An alternative approach to overcome these difficulties
  - Physics-informed neural network
  - Cyclic workflow:
    - Gaps in knowledge identified
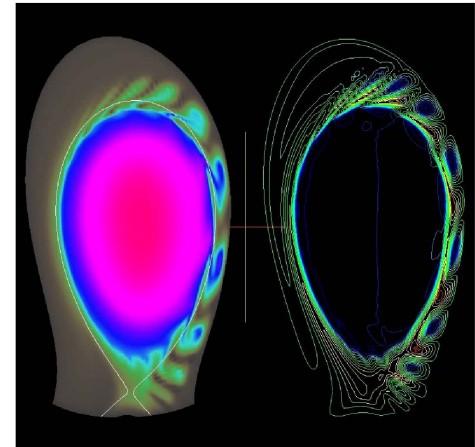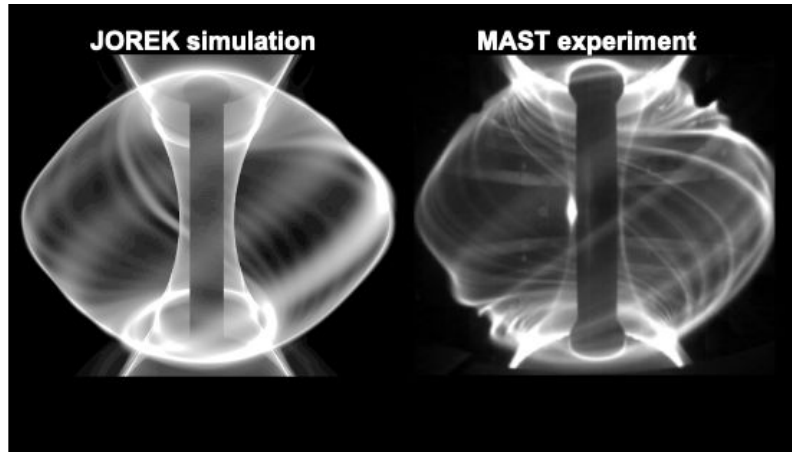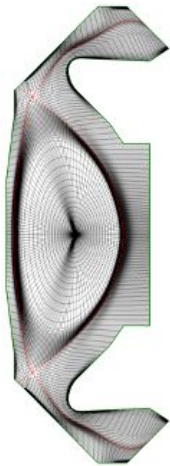    - Trigger the simulation code to generate additional data points
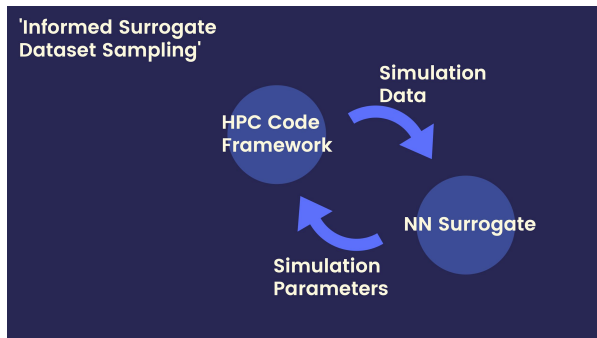


*Building NN-based surrogate models*

# Use case
*Fusion Data Space Provider*

- Fusion example: JOREK
    - Simulation of MHD instabilities at the edge of Tokamak plasmas
        - Filamentary structures ejected from the edge
        - Dynamics highly dependent on edge plasma pressure

# Technology
## *PROMINENCE*

- PROMINENCE developed in EOSCpilot & extended in EOSC-hub
- Allows users to easily exploit idle cloud resources for running scientific workloads
- Submit globally
  - Manage jobs from a web UI, batch-system like CLI, or REST API
- Run globally
  - Use any number of private, academic & public clouds simultaneously
  - Jobs are directed automatically to the most appropriate clouds
  - Infrastructure provisioning is totally invisible to users
- Reliability & reproducibility
  - All jobs are run in containers
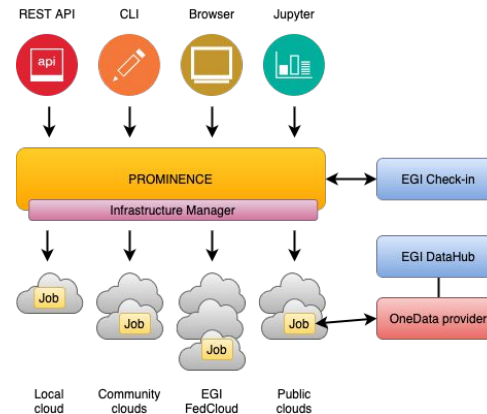


*PROMINENCE high level architecture*

# Requirement for HPC

- JOREK is a HPC code
  - For high-fidelity, requires large numbers of CPUs with low-latency interconnects
  - Typically each job would use hundreds/thousands of CPU cores across multiple nodes coupled using MPI

- On clouds with standard networking we are limited to using a single node
  - Significantly limits the fidelity
  - Significantly reduced physics
  - Results could not be used for publication - at best this becomes a 'proof of concept'
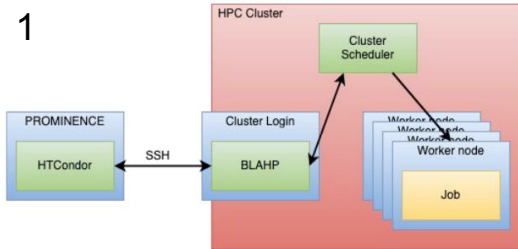
# Overall architecture

- Architecture required for the Fusion Data Space provider
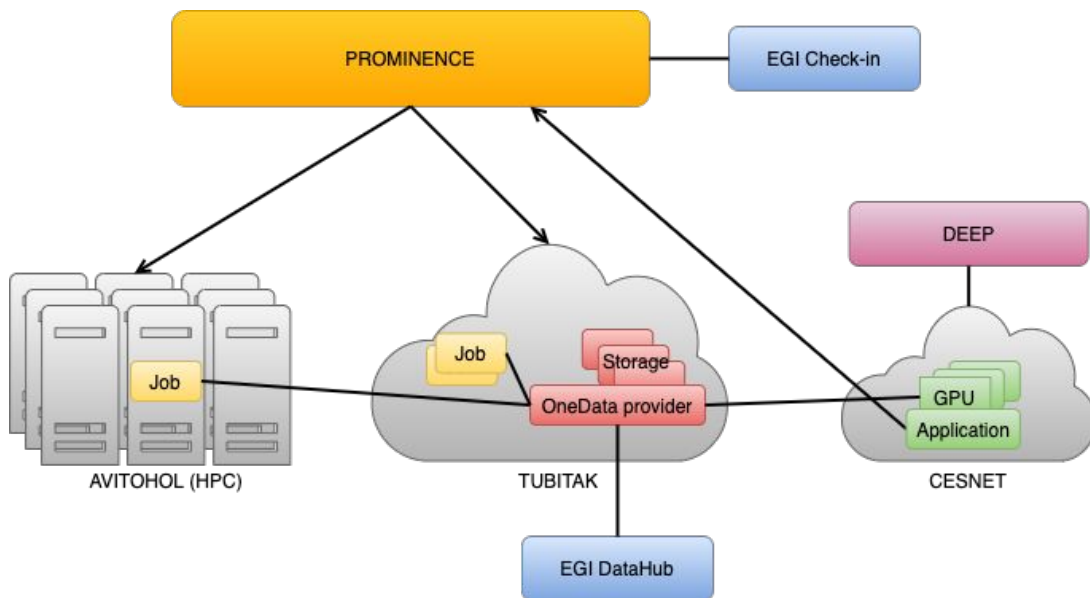
# Experience integrating UK HPC

- PROMINENCE internally uses HTCondor as the job queue & for remotely executing jobs
- Have successfully tried two methods of integrating HPC



- Vanilla universe jobs converted into Grid universe
- Jobs submitted to remote HPC cluster over ssh
- Limitation:
    - Since jobs not run on HTCondor worker nodes, lose functionality, e.g. streaming of stdout/err in real time
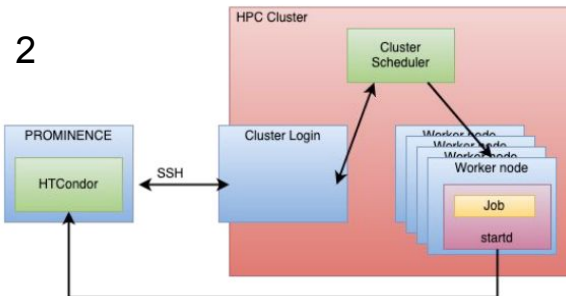


- Submit HTCondor startds (worker nodes) as jobs
    - Using RADICAL-SAGA Python module
- More consistent with how we run jobs on clouds
- Streaming of stdout/err in real time works

# Experience integrating UK HPC: challenges (1)

- We run all jobs in containers
  - Support both udocker and Singularity
- Different ways to run multi-node MPI jobs in containers on HPC
    1. MPI on host only, bind mount into containers
    2. Combination of MPI on host & MPI in containers
    3. MPI in container only
- Each have advantages & disadvantages
  - 1 & 2: need to be careful to avoid MPI version conflicts
  - 3: no version conflicts, but can be challenging to make a container which will work on multiple HPC clusters
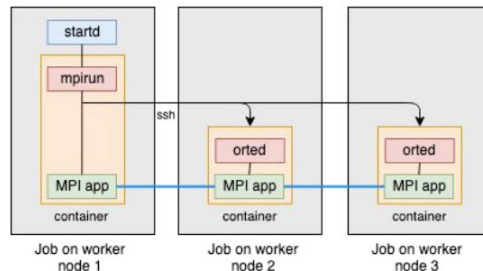


*PROMINENCE using method 3*

# Experience integrating UK HPC: challenges (2)

- Strict terms of use & policies regarding ssh keys
  - E.g. users can only store their ssh key on a single machine
  - Typically federated login not supported

- It is forbidden to have a single service submitting jobs on behalf of multiple users
  - Users can only:
    - Submit jobs via ssh manually
    - Have a cron which submits jobs
  - Related to accounting of resource usage

# Thank you!

Contact: **egi-ace-po@mailman.egi.eu**
Website: **www.egi.eu/projects/egi-ace**

**EGI Foundation**

**@EGI_eInfra**