

**ExCALIBUR   
  
NEPTUNE Charter**

**Abstract**

This document contains a set of working practices that the entire ExCALIBUR NEPTUNE  
team is expected to work to, in order to build a coherent UK wide team and to deliver  
upon the long term goals of the project – an agile, state of the art, sustainable and   
“actionable” Exascale performant and scalable platform for modelling the   
complex multi-physics of the tokamak plasma edge.

**UKAEA REFERENCE AND APPROVAL SHEET**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Client Reference: | |  | |
| UKAEA Reference: | | CD/EXCALIBUR-FMS/0013 | |
| Issue: | | 1.00 | |
| Date: | | 30 June 2020 | |
| Project Name: ExCALIBUR Fusion Modelling System.  Version 1.00 | | | | | |
|  | Name and Department | | Signature | | Date |
| Prepared By: | Wayne Arter | | N/A | | 30/06/2020 |
| Reviewed By: | Rob Akers | | N/A | | 30/06/2020 |
| Modified By: | Rob Akers  MSSC | |  | | 30/06/2020 |
| Approved By: | Rob Akers  MSSC | |  | | 30/06/2020 |

**ExCALIBUR NEPTUNE Charter**

### All members of the ExCALIBUR NEPTUNE team should be aware that to meet the challenges of the NEPTUNE project, and the ExCALIBUR overarching pillars, a distributed team of scientists, software engineers and architecture specialists from different UK institutions will be required to form a community around the NEPTUNE project (and will connect across the overarching ExCALIBUR programme). A high-level objective is to ensure that developed software is of the highest quality, implying a rigid requirement around the production of high-quality documentation and reproducible verification and validation tests for the codebase as it evolves. Since development work may transfer between institutions, it is important that common standards for documentation and testing be available and easy to deploy. The initial NEPTUNE exploratory Proxyapps may be written in a range of languages including for example python, C++/DPC++, Object Fortran or Julia, however it is envisaged that there will be an emerging steer towards a reduced set of languages and technologies to ensure interoperability across the NEPTUNE software stack, ultimately leading to coupled simulations covering all the physics necessary to deliver an “actionable” simulation for the plasma edge. It is not yet clear for example whether SYCL, Kokkos or OpenMP 5 will offer the most performance portable and sustainable solution for NEPTUNE, making down selection difficult at this point in time. The team is therefore expected to be agile and amenable to change once it is clear which are the most promising long-term solutions. For example, a selection of SYCL for the long-term framework/code(s) would force refactoring of any code that is not consistent with a NEPTUNE library and code base instantiated in Data Parallel C++, and where feasible, team members should support this process

### Source code for all development should be accessible by the entire NEPTUNE team and all tests should be repeatable by different workers without the need for re-training and/or any possible confusion as to the procedures and metrics needed to declare a test successful.

NEPTUNE will be developed as a sequence of ‘core’ Proxyapps (to be distinguished from other Proxyapps designed to test some novel technique). Core Proxyapps will all need a documentation and testing framework, which must be agreed between all partners for the entire project. This will require developers to work closely with UKAEA and other team members.

A commitment is also expected by all parties to help UKAEA and the Met Office (as SRO for ExCALIBUR) to publicise the project and build a fully connected community across the ExCALIBUR programme, UKRI and Academia, focused upon a team of approximately 20 UK Fusion use case experts. This will be essential for meeting the grand challenge goal of developing a state-of-the-art, Exascale targeted, UK based plasma physics simulation capability for the tokamak plasma “edge” (see Science Plan [1]).

All Core Proxyapps and related infrastructure/documentation across the NEPTUNE project should meet the demands of the Code Structure and Coordination work package FM-WP4 in so far as the developing project standards:

* adopt a consistent choice of definitions (ontology) of objects or equivalently classes,
* adhere to clearly defined common file formats and interfaces to components for data IO,
* provide suitably flexible data structures for common use by all developers,
* are established through good scientific software engineering best practice,
* demonstrate performance portability and exploit agreed DSL like interfaces where possible targeting Exascale-relevant architectures,
* can be integrated into a VVUQ framework and
* are embedded within a coordination and benchmarking framework for correctness testing and performance evaluation.

# References

[1] W. Arter, L. Anton, D. Samaddar, and R. Akers. ExCALIBUR Fusion Modelling System Science Plan. Technical Report CD/EXCALIBUR-FMS/0001, UKAEA, 2019.