# T-NA083 Development of fluid referent models

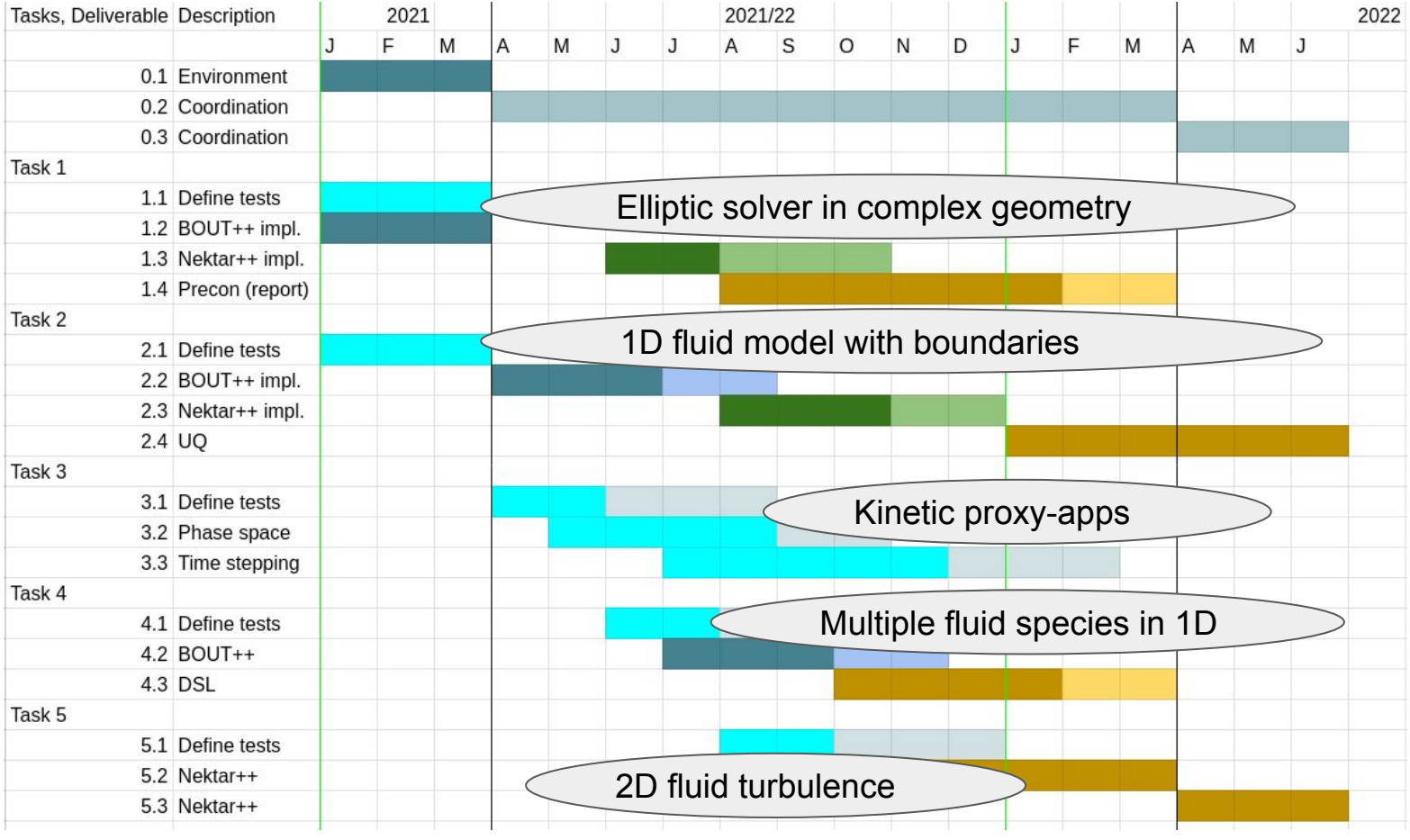
B.Dudson[1], D.Moxey[2], S.Wright[1], P.Hill[1], D.Dickinson[1], E.Higgins[1]

[1] University of York
[2] University of Exeter

# Aims

1. Bring together and integrate the work of other work packages
2. Develop increasingly complicated test cases, with reference and spectral/hp implementations, in steps towards the full NEPTUNE fusion simulation
3. Write exploratory proxy-apps to test algorithms and strategies
4. Coordinate code development practices and training

# Project timeline



# Task 5, system 2.2

Moderately complex plasma model in 2D. Simplified compared to full system!

$$\partial_t n_e + \nabla \cdot (n_e \mathbf{u}_e) = S_{n_e} - \frac{n_e}{\tau_{n_e}}$$

$$\partial_t \nabla \cdot \mathbf{E}^+ + \nabla \cdot \left(\nabla \cdot \left(\mathbf{u}_i \otimes \mathbf{E}^+\right)\right) = \nabla \cdot \left( n_i \left(\mathbf{u}_{\nabla Bi} + \mathbf{u}_{cx}\right) - \frac{1}{Z_i} n_e \mathbf{u}_{\nabla Be} \right) + \frac{1}{Z_i} \frac{n_e}{\tau_{n_e}} - \frac{n_i}{\tau_{n_i}} + \nabla \cdot \left(\nu \nabla_\perp \left(\nabla \cdot \mathbf{E}^+\right)\right)$$

$$\partial_t \mathcal{E}_e + \nabla \cdot (\mathcal{E}_e \mathbf{u}_e + p_e \mathbf{u}_e) = S_{\mathcal{E}_e} - \frac{\mathcal{E}_e}{\tau_{Ee}} + Q_{ie} + \nabla \cdot (\chi_{\perp e} n_e \nabla_\perp T_e)$$

$$\partial_t \mathcal{E}_i + \nabla \cdot (\mathcal{E}_i \mathbf{u}_i + p_i \mathbf{u}_i) = S_{\mathcal{E}_i} - \frac{\mathcal{E}_i}{\tau_{Ei}} - Q_{ie} + \nabla \cdot (\chi_{\perp i} n_i \nabla_\perp T_i)$$

$$\partial_t n_n = S_{n_n} + \nabla \cdot (D_n \nabla_\perp p_n)$$

# Task 5, system 2.2

Moderately complex plasma model in 2D. Simplified compared to full system!

$$\partial_t n_e + \nabla \cdot (n_e \mathbf{u}_e) = S_{n_e} - \frac{n_e}{\tau_{n_e}}$$

$$\partial_t \nabla \cdot \mathbf{E}^+ + \nabla \cdot \left(\nabla \cdot \left(\mathbf{u}_i \otimes \mathbf{E}^+\right)\right) = \nabla \cdot \left( n_i \left(\mathbf{u}_{\nabla Bi} + \mathbf{u}_{cx}\right) - \frac{1}{Z_i} n_e \mathbf{u}_{\nabla Be} \right)$$

All coefficients nonlinear, time-dependent

$$+ \frac{1}{Z_i} \frac{n_e}{\tau_{n_e}} - \frac{n_i}{\tau_{n_i}} + \nabla \cdot \left(\nu \nabla_\perp \left(\nabla \cdot \mathbf{E}^+\right)\right)$$

$$\partial_t \mathcal{E}_e + \nabla \cdot (\mathcal{E}_e \mathbf{u}_e + p_e \mathbf{u}_e) = S_{\mathcal{E}_e} - \frac{\mathcal{E}_e}{\tau_{Ee}} + Q_{ie} + \nabla \cdot (\chi_{\perp e} n_e \nabla_\perp T_e)$$

$$\partial_t \mathcal{E}_i + \nabla \cdot (\mathcal{E}_i \mathbf{u}_i + p_i \mathbf{u}_i) = S_{\mathcal{E}_i} - \frac{\mathcal{E}_i}{\tau_{Ei}} - Q_{ie} + \nabla \cdot (\chi_{\perp i} n_i \nabla_\perp T_i)$$

$$\partial_t n_n = S_{n_n} + \nabla \cdot (D_n \nabla_\perp p_n)$$

# Task 5, system 2.2

Moderately complex plasma model in 2D. Simplified compared to full system!

$$\partial_t n_e + \nabla \cdot (n_e \mathbf{u}_e) = S_{n_e} - \frac{n_e}{\tau_{n_e}}$$

$$\partial_t \nabla \cdot \mathbf{E}^+ + \nabla \cdot \left(\nabla \cdot \left(\mathbf{u}_i \otimes \mathbf{E}^+\right)\right) = \nabla \cdot \left( n_i \left(\mathbf{u}_{\nabla Bi} + \mathbf{u}_{cx}\right) - \frac{1}{Z_i} n_e \mathbf{u}_{\nabla Be} \right)$$

$$+ \frac{1}{Z_i} \frac{n_e}{\tau_{n_e}} - \frac{n_i}{\tau_{n_i}} + \nabla \cdot \left(\nu \nabla_\perp \left(\nabla \cdot \mathbf{E}^+\right)\right)$$

Many coefficients are tensors, highly anisotropic

FE representation

$$\partial_t \mathcal{E}_e + \nabla \cdot (\mathcal{E}_e \mathbf{u}_e + p_e \mathbf{u}_e) = S_{\mathcal{E}_e} - \frac{\mathcal{E}_e}{\tau_{Ee}} + Q_{ie} + \nabla \cdot (\chi_{\perp e} n_e \nabla_\perp T_e)$$

$$\partial_t \mathcal{E}_i + \nabla \cdot (\mathcal{E}_i \mathbf{u}_i + p_i \mathbf{u}_i) = S_{\mathcal{E}_i} - \frac{\mathcal{E}_i}{\tau_{Ei}} - Q_{ie} + \nabla \cdot (\chi_{\perp i} n_i \nabla_\perp T_i)$$

$$\partial_t n_n = S_{n_n} + \nabla \cdot (D_n \nabla_\perp p_n)$$

# Task 5, system 2.2

Moderately complex plasma model in 2D. Simplified compared to full system!

$$\partial_t n_e + \nabla \cdot (n_e \mathbf{u}_e) = S_{n_e} - \frac{n_e}{\tau_{n_e}}$$

$$\partial_t \nabla \cdot \mathbf{E}^+ + \nabla \cdot \left(\nabla \cdot \left(\mathbf{u}_i \otimes \mathbf{E}^+\right)\right) = \nabla \cdot \left( n_i \left(\mathbf{u}_{\nabla Bi} + \mathbf{u}_{cx}\right) - \frac{1}{Z_i} n_e \mathbf{u}_{\nabla Be} \right)$$

$$\mathbf{E}^+ = \frac{m_i}{Z_i e B^2} \left( n_i \nabla_\perp \Phi + \frac{1}{Z_i e} \nabla_\perp p_i \right)$$

$+ \nabla \cdot \left(\nu \nabla_\perp \left(\nabla \cdot \mathbf{E}^+\right)\right)$

4th derivative of $\Phi$

$+ \nabla \cdot (\chi_{\perp e} n_e \nabla_\perp T_e)$

$\nabla \cdot (\chi_{\perp i} n_i \nabla_\perp T_i)$

Advection velocities **u** depend on $\nabla \Phi$

Vorticity equation, elliptic inversion (global) with time-varying coefficients an important part of the problem

Preconditioning

# Task 5, system 2.2

Moderately complex plasma model in 2D. Simplified compared to full system!

$$\partial_t n_e + \nabla \cdot (n_e \mathbf{u}_e) = S_{n_e} - \frac{n_e}{\tau_{n_e}}$$

$$\partial_t \nabla \cdot \mathbf{E}^+ + \nabla \cdot \left(\nabla \cdot \left(\mathbf{u}_i \otimes \mathbf{E}^+\right)\right) = \nabla \cdot \left( n_i \left(\mathbf{u}_{\nabla Bi} + \mathbf{u}_{cx}\right) - \frac{1}{Z_i} n_e \mathbf{u}_{\nabla Be} \right)$$

$$+ \nabla \cdot \left(\nu \nabla_\perp \left(\nabla \cdot \mathbf{E}^+\right)\right)$$

Sources from particle model, reaction rates with uncertainties

Uncertainty Quantification

Coupling to particles

$$\partial_t \mathcal{E}_e + \nabla \cdot (\mathcal{E}_e \mathbf{u}_e + p_e \mathbf{u}_e) = S_{\mathcal{E}_e} - \frac{\mathcal{E}_e}{\tau_{Ee}} + Q_{ie} + \nabla \cdot (\chi_{\perp e} n_e \nabla_\perp T_e)$$

$$\partial_t \mathcal{E}_i + \nabla \cdot (\mathcal{E}_i \mathbf{u}_i + p_i \mathbf{u}_i) = S_{\mathcal{E}_i} - \frac{\mathcal{E}_i}{\tau_{Ei}} - Q_{ie} + \nabla \cdot (\chi_{\perp i} n_i \nabla_\perp T_i)$$

$$\partial_t n_n = S_{n_n} + \nabla \cdot (D_n \nabla_\perp p_n)$$

# Task 5, system 2.2

Moderately complex plasma model in 2D. Simplified compared to full system!

$$\partial_t n_e + \nabla \cdot (n_e \mathbf{u}_e) = S_{n_e} - \frac{n_e}{\tau_{n_e}}$$

$$\partial_t \nabla \cdot \mathbf{E}^+ + \nabla \cdot \left(\nabla \cdot \left(\mathbf{u}_i \otimes \mathbf{E}^+\right)\right) = \nabla \cdot \left(n_i \left(\mathbf{u}_{\nabla Bi} + \mathbf{u}_{cx}\right) - \frac{1}{Z_i} n_e \mathbf{u}_{\nabla Be}\right)$$

This is only one ion species
NEPTUNE must have many

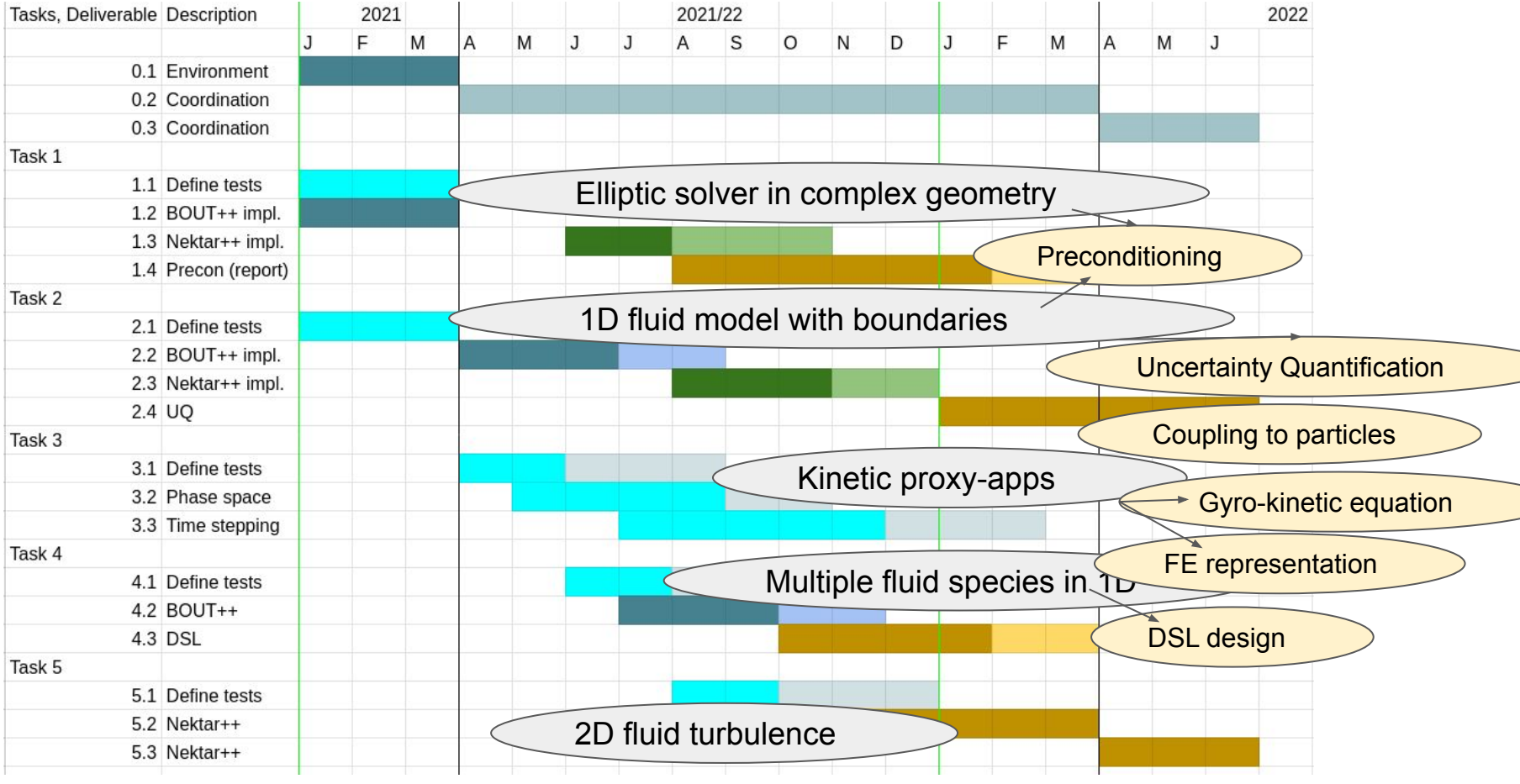
DSL design

$$\partial_t \mathcal{E}_e + \nabla \cdot (\mathcal{E}_e \mathbf{u}_e + p_e \mathbf{u}_e) = S_{\mathcal{E}_e} - \frac{\mathcal{E}_e}{\tau_{Ee}} + Q_{ie} + \nabla \cdot (\chi_{\perp e} n_e \nabla_\perp T_e)$$

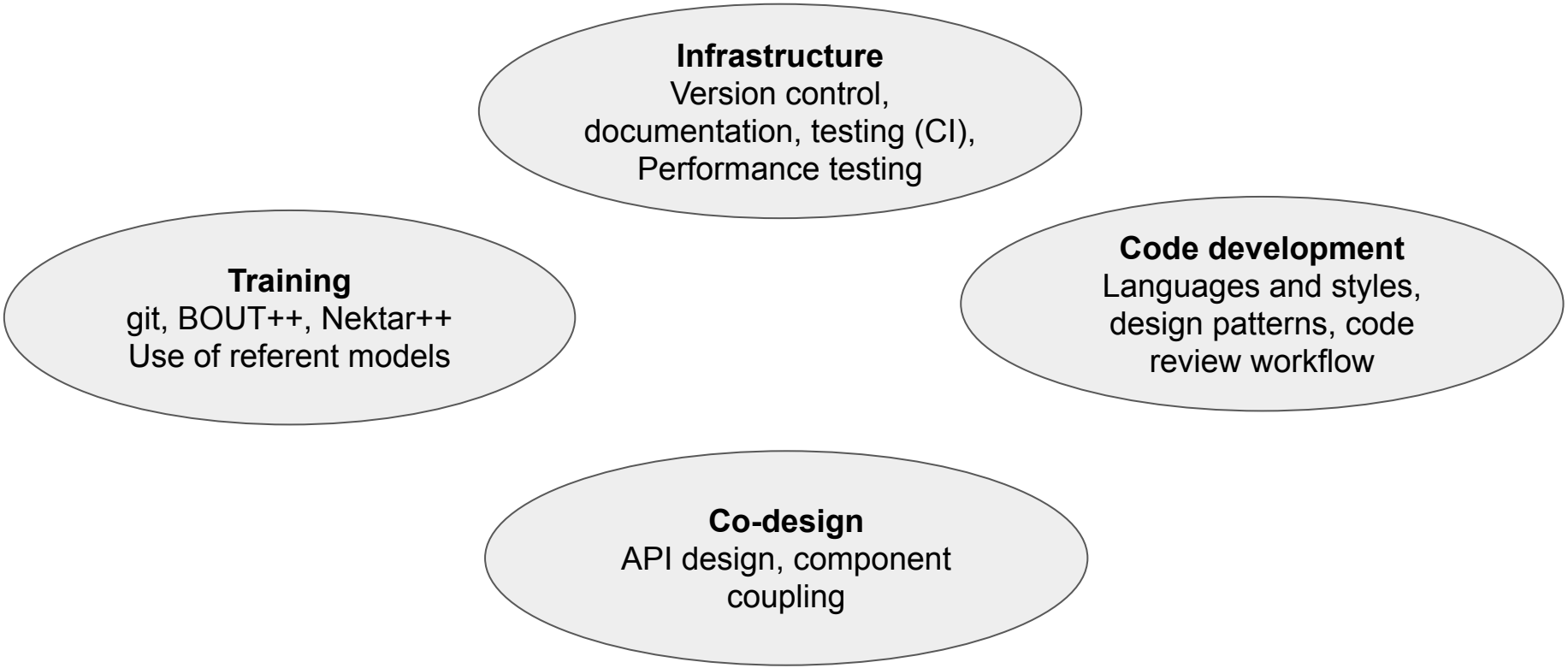
$$\partial_t \mathcal{E}_i + \nabla \cdot (\mathcal{E}_i \mathbf{u}_i + p_i \mathbf{u}_i) = S_{\mathcal{E}_i} - \frac{\mathcal{E}_i}{\tau_{Ei}} - Q_{ie} + \nabla \cdot (\chi_{\perp i} n_i \nabla_\perp T_i)$$

$$\partial_t n_n = S_{n_n} + \nabla \cdot (D_n \nabla_\perp p_n)$$

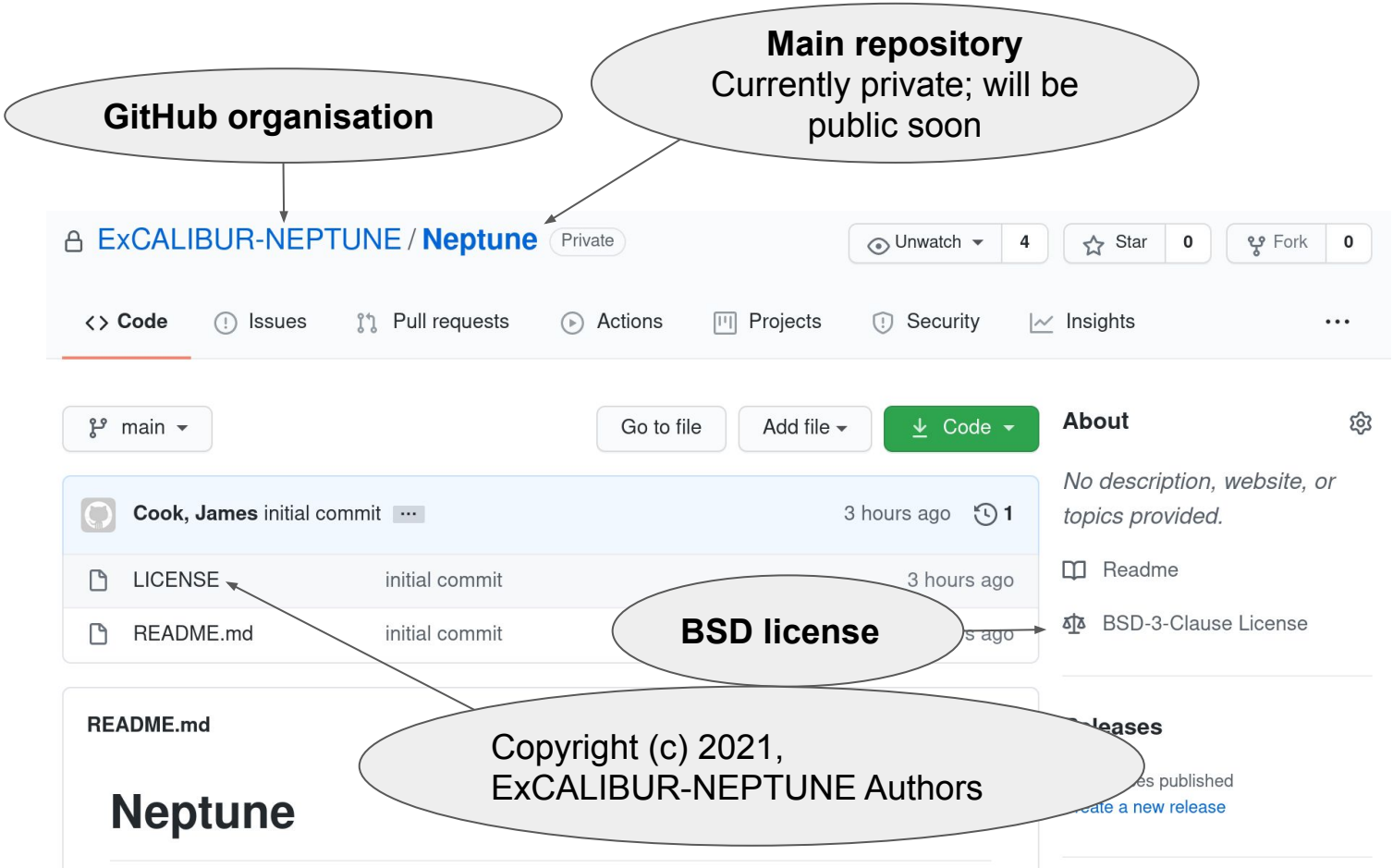
# Integration with other work packages



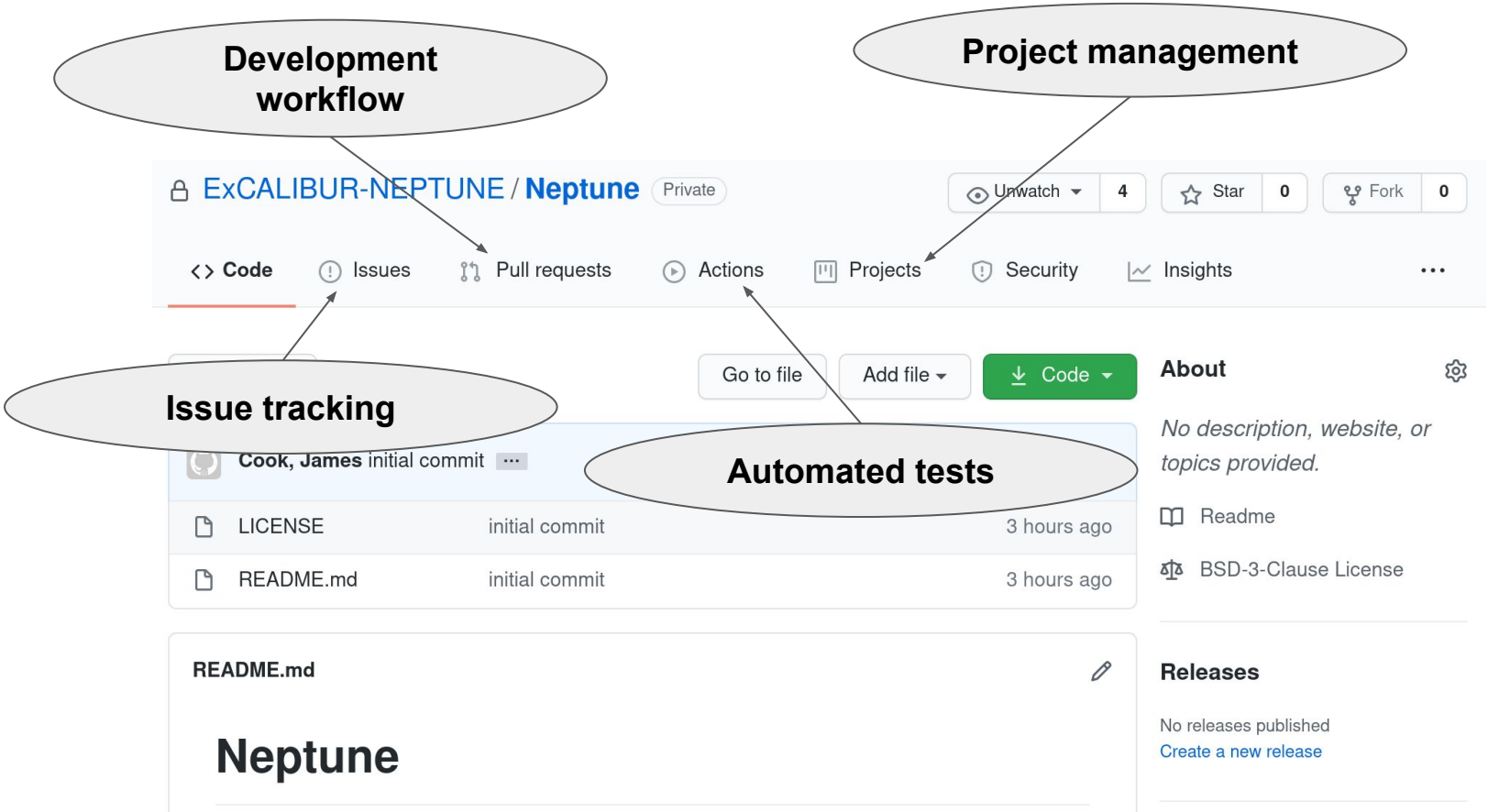
# Community coordination



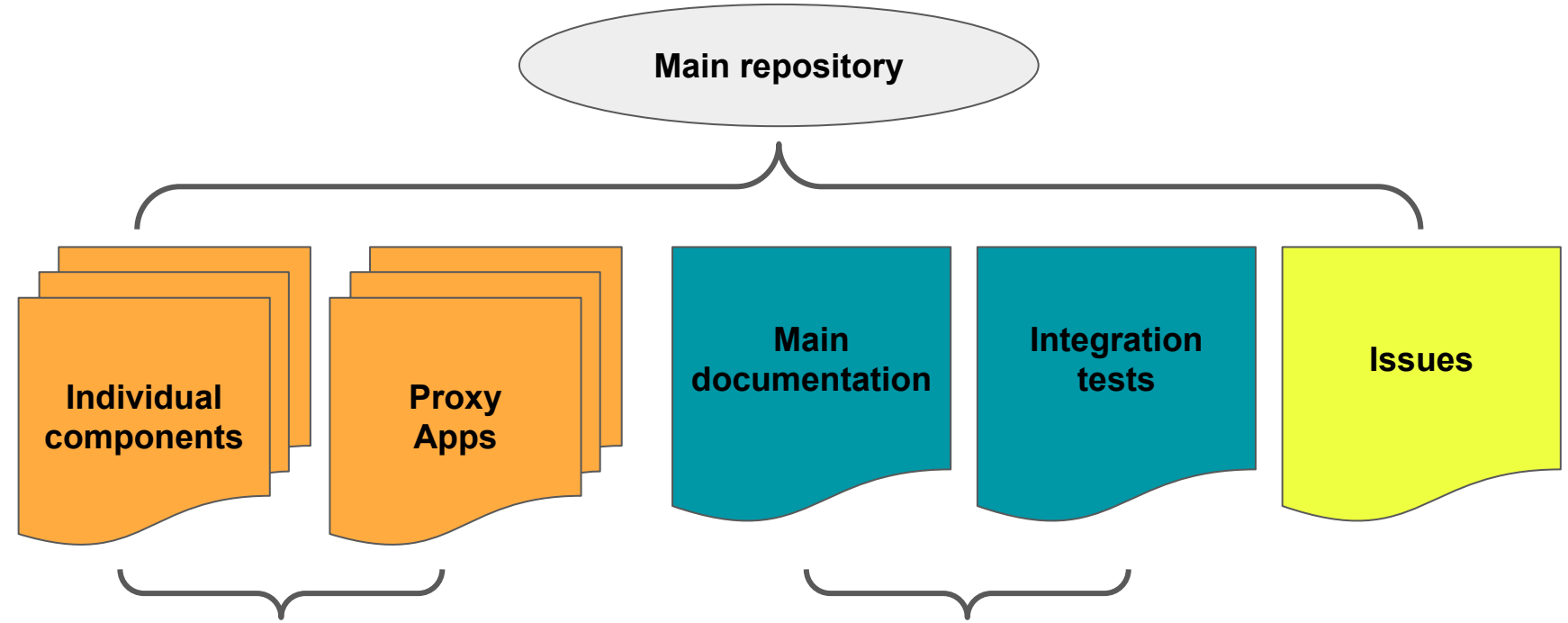
# Version control: GitHub



# Version control: GitHub

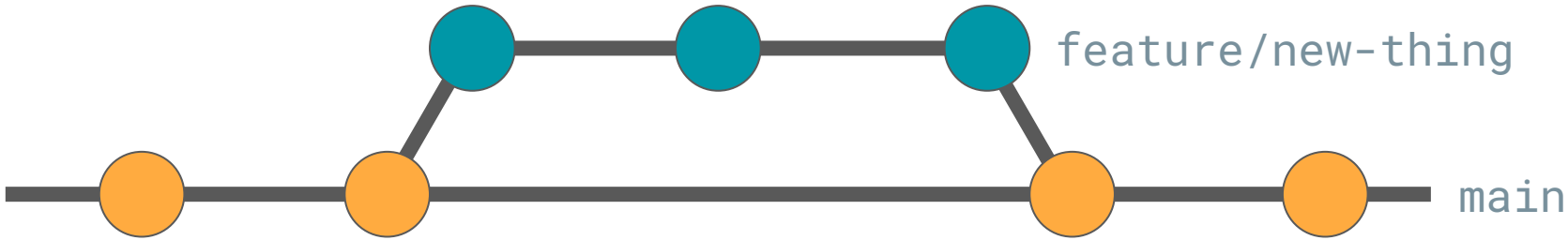


# Version control: Organisation



# Version control: Workflow

- Development to be done via feature branches and pull requests (PRs)
- Allows for code review and discussion — vital for developing cohesive project
  - code review is a responsibility of all developers
- Larger components may have additional `develop` branch
- Automated tests run on PRs — must pass before merging



# Testing

- Individual components contain their own tests
- Write tests as early as possible (even before the code itself!)
- Integration/coupling tests in main repo
  - Will ensure the various components work together
- GitHub Actions to automatically run tests/static analysis on commits/PRs
- Performance testing: likely to be semi-automated, at least for time being
  - Will need to run on variety of hardware/environments
  - Data schema likely to evolve

# Code development

- Development to all be done in public repos
- Common licence: BSD-3
  - “Do what you want, but keep attribution and licence”
- Code style:
  - Use automated tools wherever possible (`clang-format`, `black`)
  - Exact style less important than consistency (but see above)
- Documentation:
  - Use of “doc-strings”, in-line documentation
  - Doxygen + breathe + sphinx + ReadTheDocs == automated building of documentation
  - Markdown/reStructuredText: both allow embedded LaTeX for equations
- Programming languages:
  - (Try to) use latest standards for new code: C++20, Python 3.9, Fortran 2018
    - Limited by availability on machines, but e.g. ARCHER2 has gcc 10.1