FORGE: Re-generating a forecast system for sustainability

21st ECMWF workshop on high performance computing in meteorology

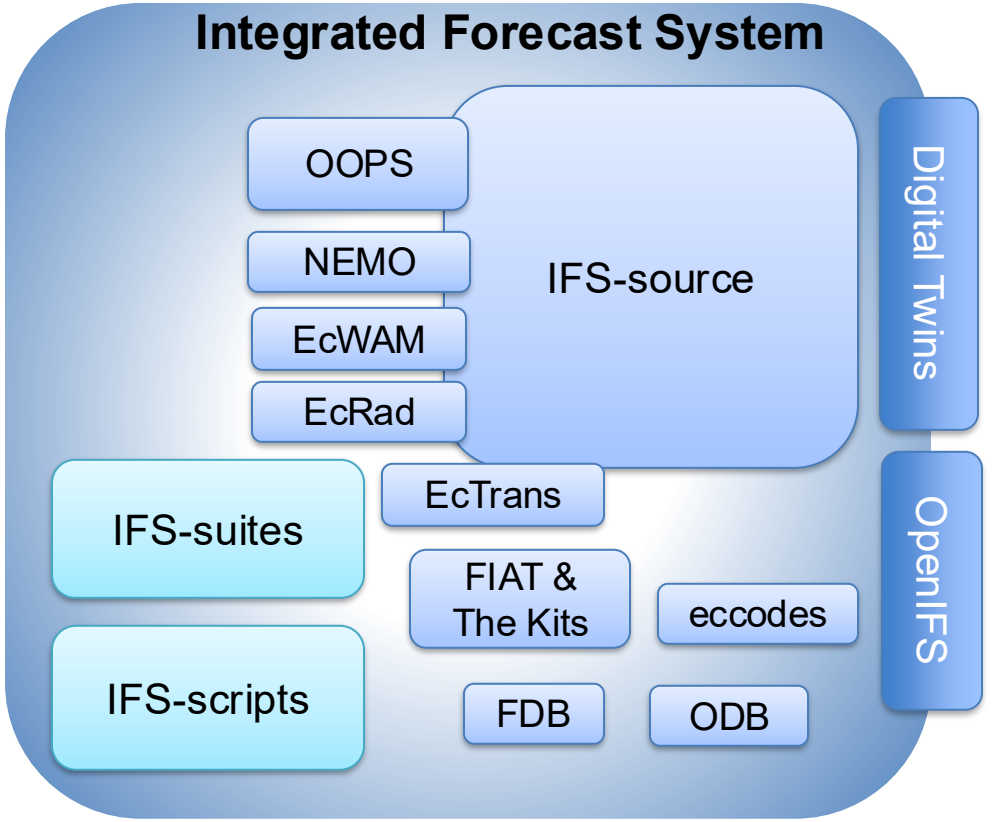
Michael Sleigh, **Michael Lange** (presenting), Andrew Bennett, Paul Burton, Zak Kipling, Ahmad Nawab

ECMWF

FORGE

The IFS today

- The IFS is an extremely complex system
  - Development driven by meteorological quality
  - Large range of applications and expanding further
    - 4DVar, ensemble forecasting, ensemble DA
    - Extended and seasonal, atmosphere/ocean/land reanalysis
    - Atmospheric composition and GHG forecasting/reanalysis
    - Flood and fire forecasting, extreme weather and climate
- Need to adapt to a changing environment
  - External data centres and HPC machines (with GPUs)
  - AIFS, data-driven methods and hybrid ML



The IFS today

- The challenge
  - Motivation, scope – what are we doing, and why
  - Technical debt / entropy in software
  - Our forecasting systems
  - Computing landscape / environment, new methods

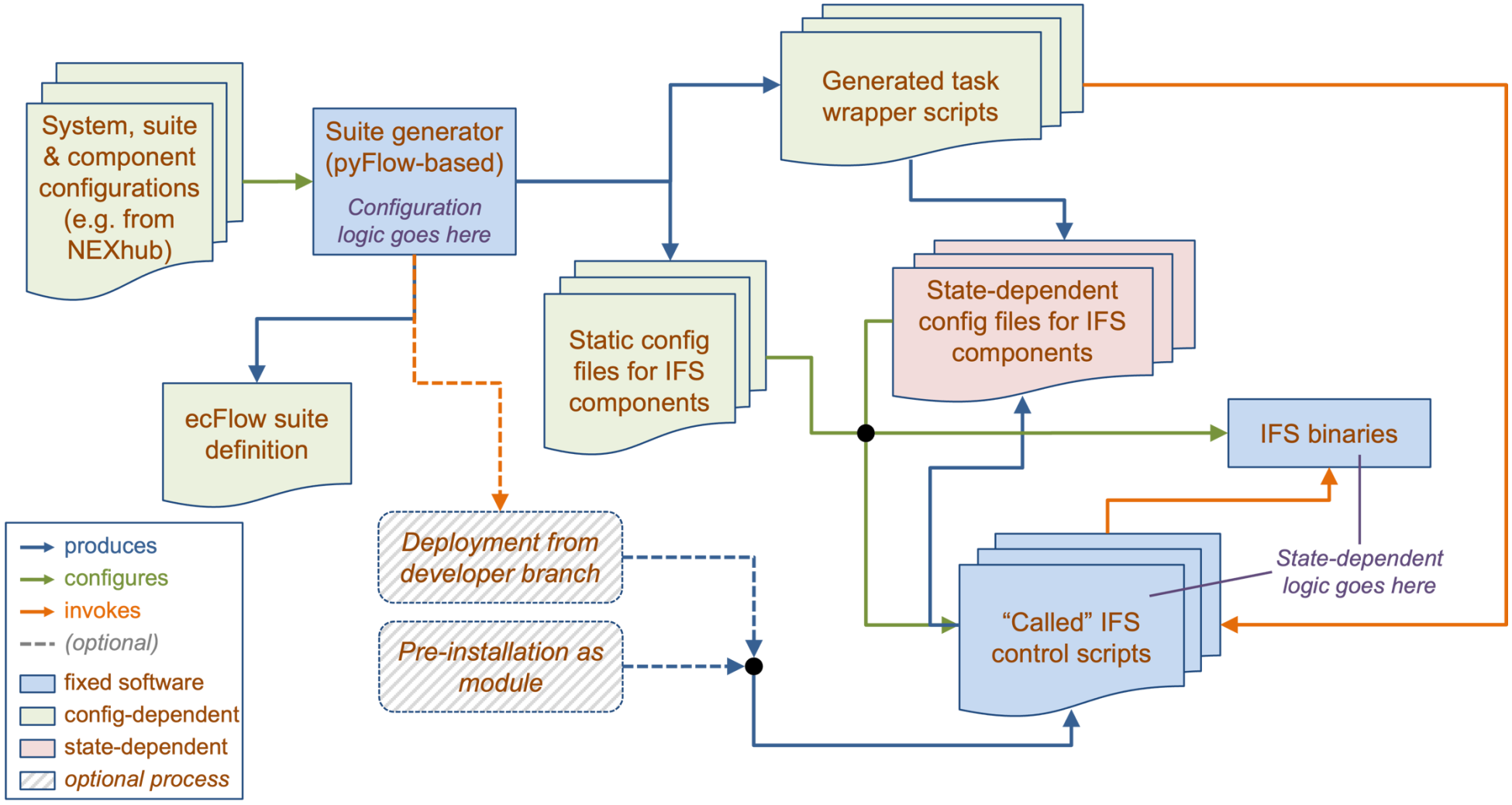
- The strategy
  - Software engineering principles
  - Separation of concerns / encapsulation
  - Testing
  - Added value to Member States

ECMWF FORGE

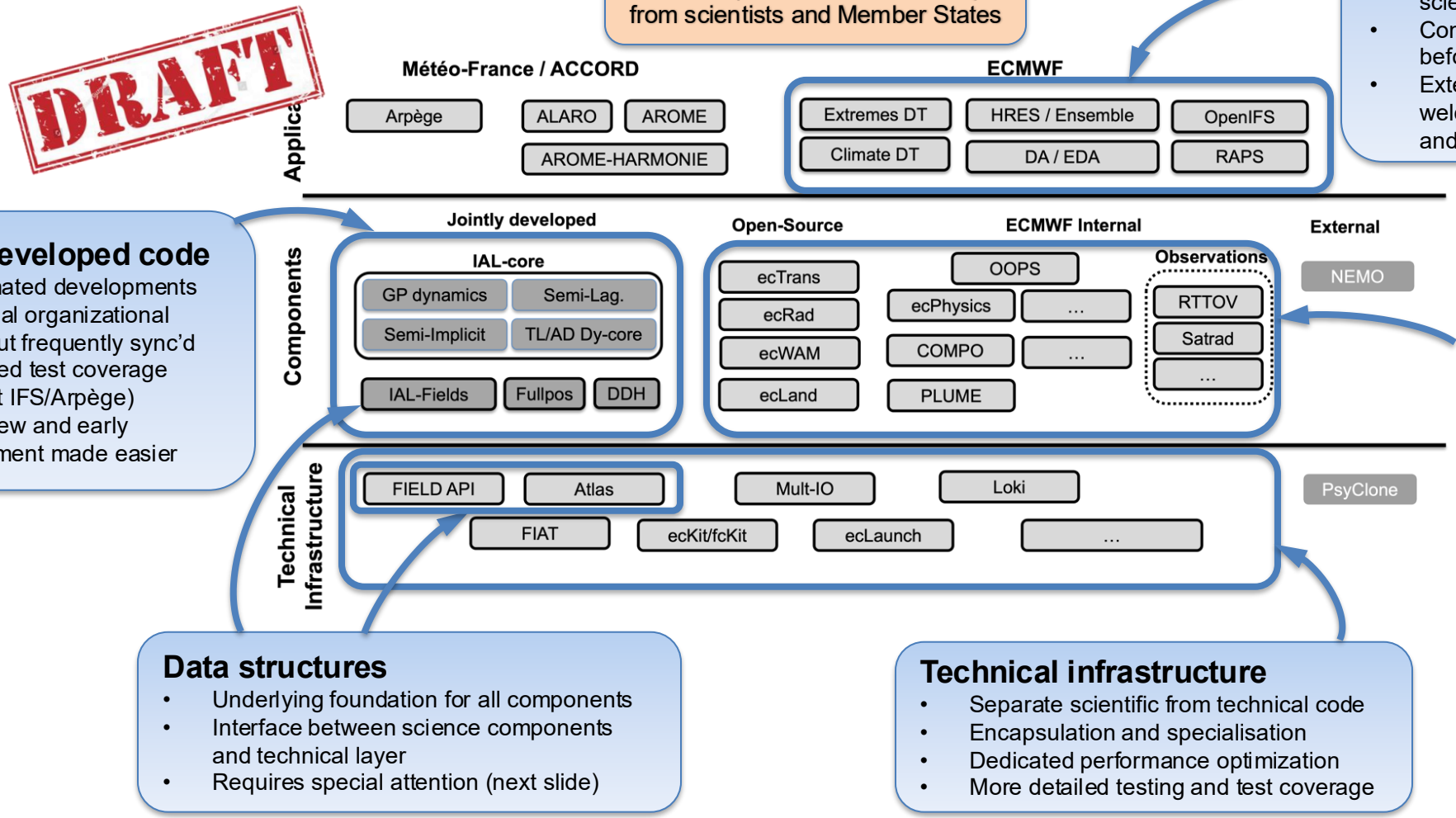
Forecast System Regeneration

- A major initiative to regenerate the IFS code
- Five-year timeframe, 2025-2029
- Covering:
  - **WP1: Workflow Code**
  - **WP2: Model Code Refactoring**
  - **WP3: Object-oriented data structures**
  - **WP4: Infrastructure**
  - **WP5: Processes and Workflows**

IFS Workflow Code: overview



Proposed structure of the compiled IFS code base



WARNING: This is a draft!
We will improve contours with input from scientists and Member States

Release IFS cycles

- No change to nature of scientific cycle releases!
- Component version tested before integration
- External contributions welcome (PRs), but vetted and tested by ECMWF

Jointly developed code

- Coordinated developments
- Individual organizational forks, but frequently sync'd
- Increased test coverage (not just IFS/Arpège)
- PR review and early assessment made easier

ECMWF-owned code

- ECMWF manages code
- Open-source or closed source (but open to MS)
- Clean interfaces for external use; tested(!)
- External contributions welcome (PRs), but vetted and tested by ECMWF

Data structures

- Underlying foundation for all components
- Interface between science components and technical layer
- Requires special attention (next slide)

Technical infrastructure

- Separate scientific from technical code
- Encapsulation and specialisation
- Dedicated performance optimization
- More detailed testing and test coverage

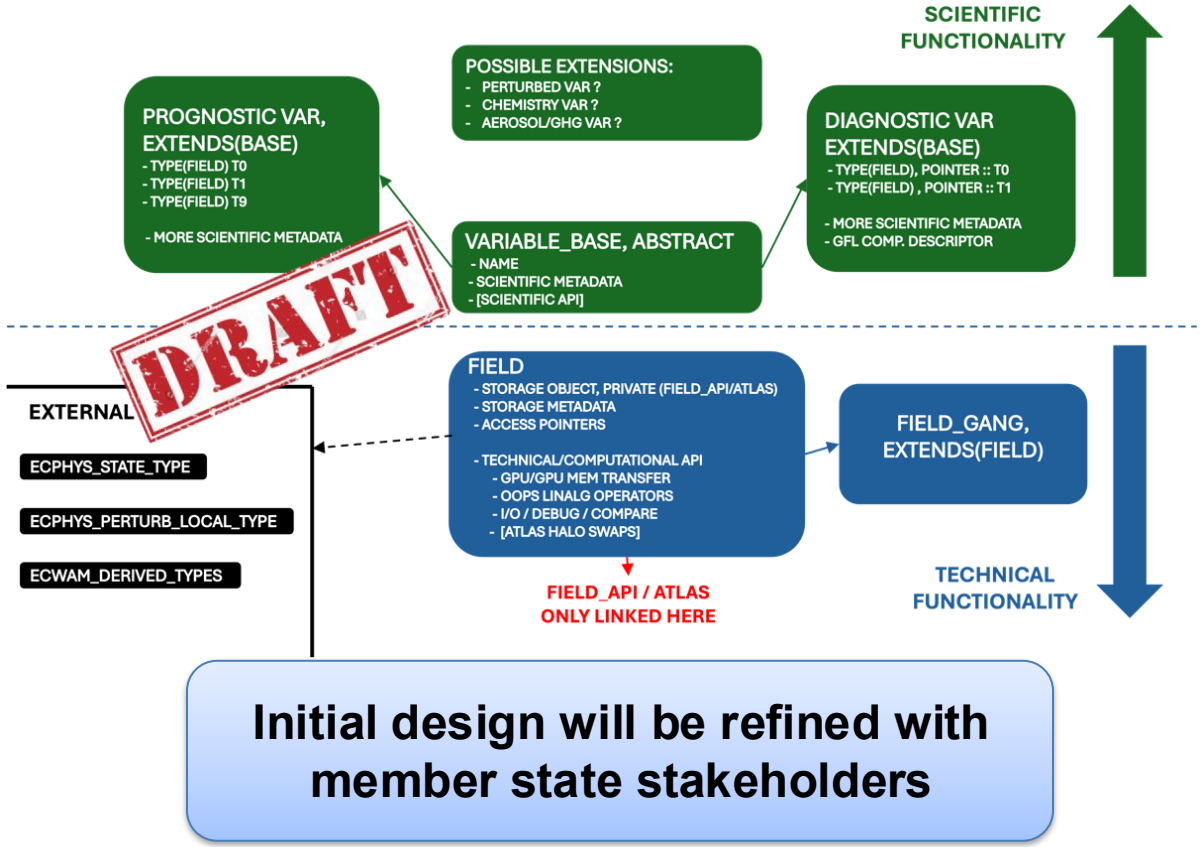
ECMWF FORGE

IFS data structures – FIELD/VARIABLE abstraction layer

Shared data layout across IFS-Arpège
- Object-oriented field abstractions
  - Technical extensions for GPU porting
- Used in shared and non-shared modules

Object-oriented data structures for science
- Recent addition of Field-API was technical
- How can we improve the science coding?
  - Shared utilities across modules
  - Robustness and testing

Atlas as an abstraction for grid data
- Field interpolation with GPU-support
- Support for many grid types



NEXHUB

- Largely invisible to users
- Provides services to plugin apps
- Databases for apps
- https API

ExpERT

Experiment Edit & Run Tool
(prepIFS replacement)

- Find experiments (all users)
- Copy & Create experiments
- Edit experiments
- Run macros
- Rule based updates (checks)
- Compare experiments
- Submit experiments

DLM

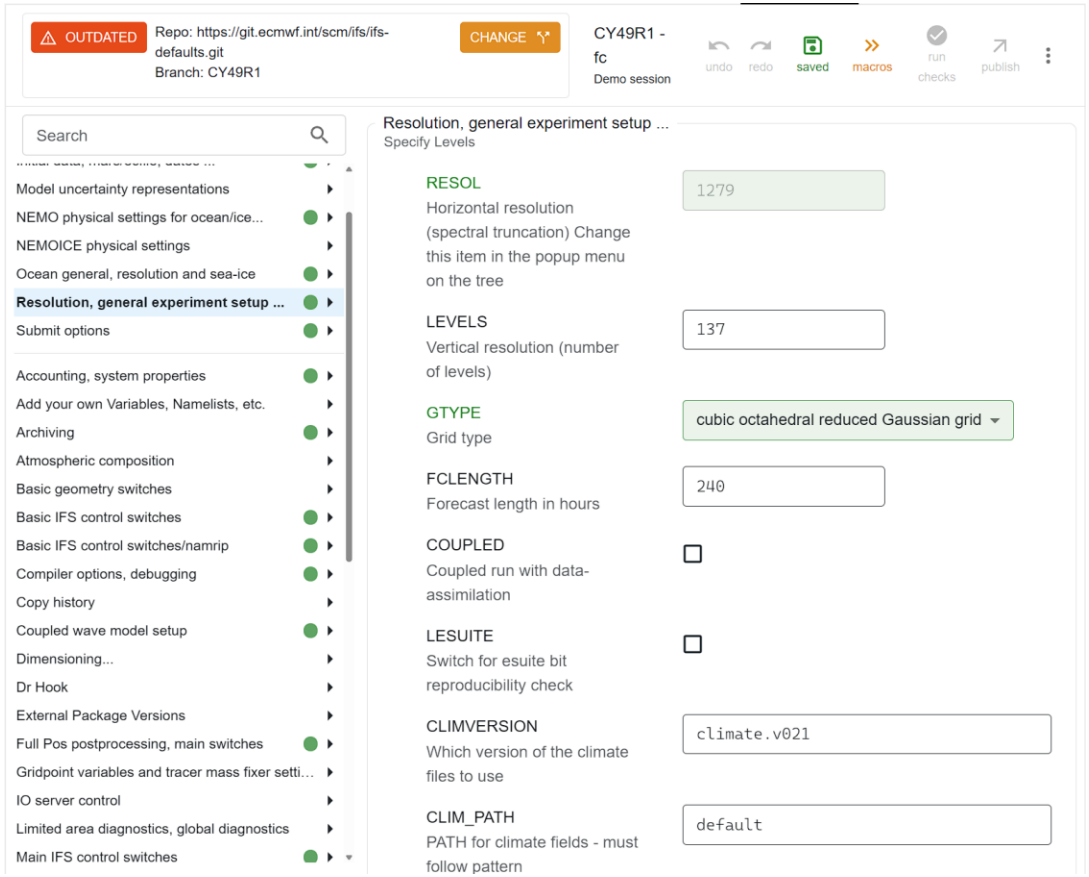
Quaver | IVER | Plots | Performance

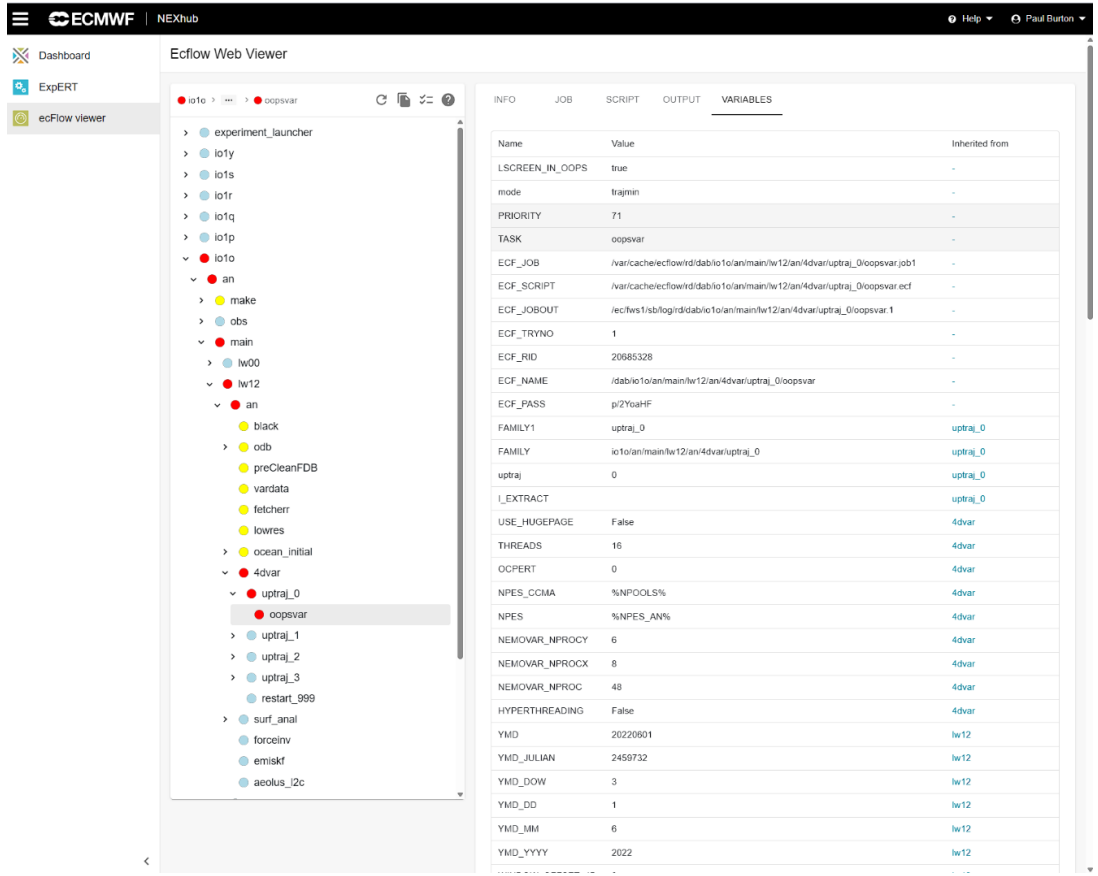
ecFlow View

ecFlow Viewer

- Limited functionality
- Shows state of experiment (read only)
- Tree-based view, no customisation

NEXhub

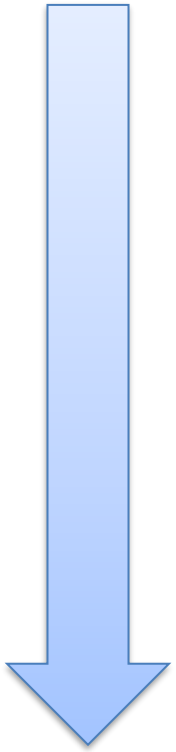




ECMWF FORGE

A multi-layered approach to testing

Modular, targeted



System, end-to-end

- Multiple levels of testing:

NEW

- **Unit/modular testing**
  - **Testing modularised components of Fortran/C/C++ source, shell scripts and ecFlow suites**
- Targeted configuration testing - manually triggered
  - Git ifstest, IFS experiments
- Automated
  - Automatically triggered on pull-requests or events
- Continuous *Integration* (CI)
  - IFS citest (running standard set of experiments, if all succeed automatically merging to master)

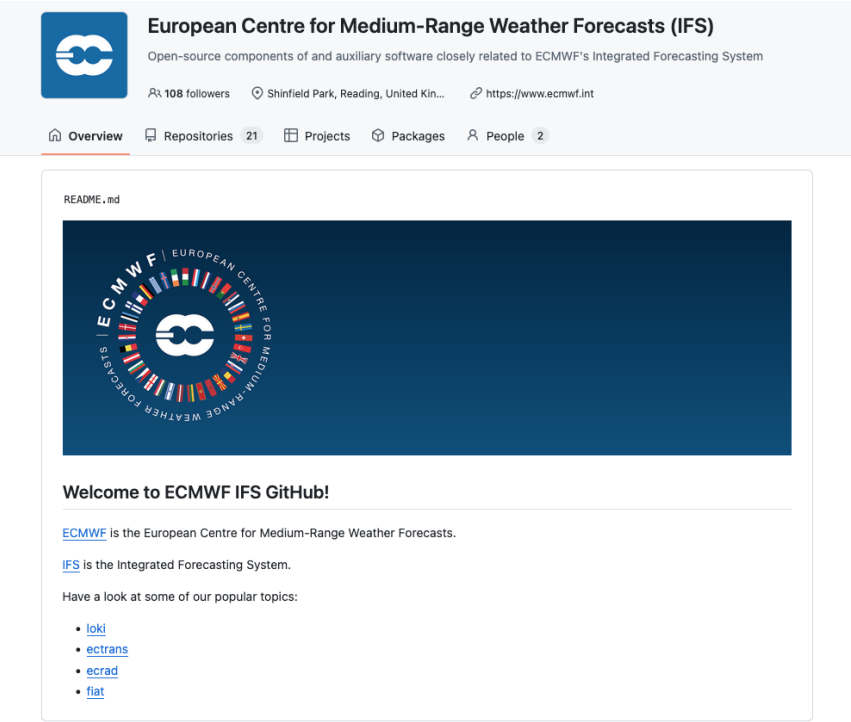
NEW

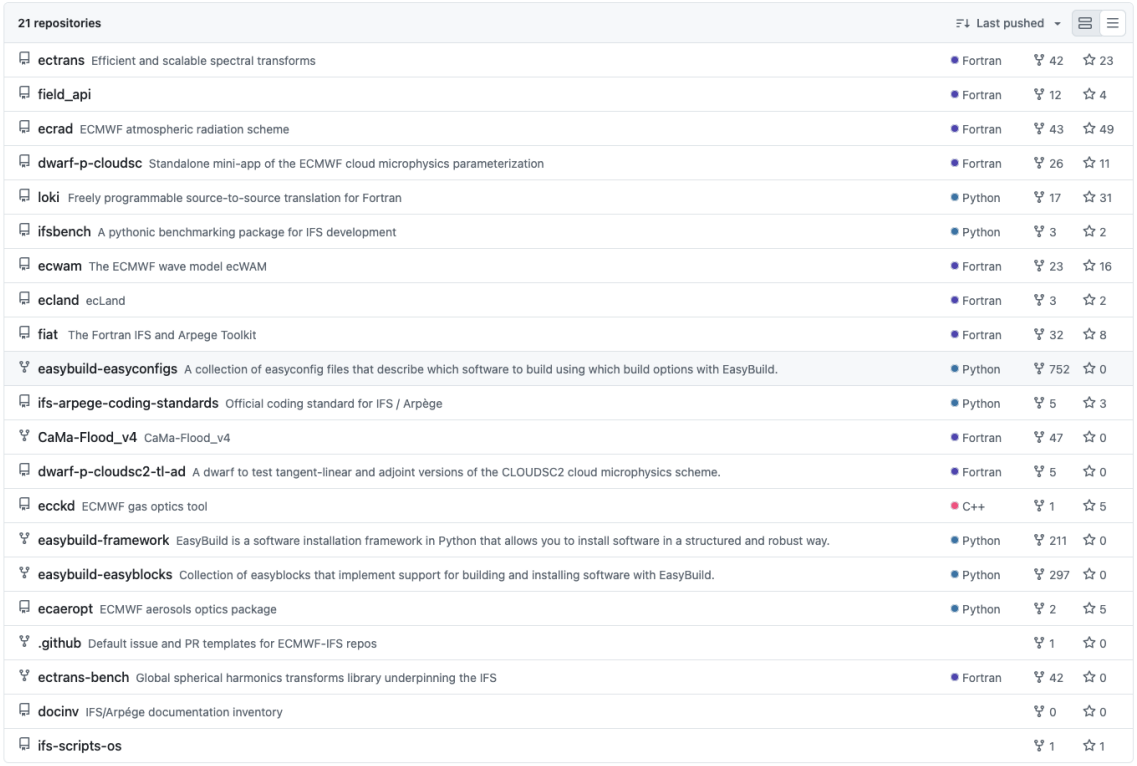
- **Continuous *Deployment* (CD)**
  - **Completely automated process from testing through to creating a release and deployment to D-suite/production (with appropriate human approval points)**
- **D-suite**
  - **An end-to-end clone of the operational suite, running in real-time**

ECMWF FORGE

Open collaborative development

- Since 2022 have been making components of IFS open-source under Apache-2 licence
- ECMWF strategy states: "… *in the spirit of 'open science' and … to further collaboration (e.g. with academia) and better serve the scientific community, ECMWF will build on the successful OpenIFS efforts and move to an open-source approach for the whole of the forecast model*"





ECMWF FORGE

FORGE - Forecast System Regeneration

- **A major initiative to regenerate the IFS code**
  - Tackling technical debt over the coming 5 years
  - Restructure model code and data structures
  - Modernize the workflows supporting software stack

- **Increased resilience in a changing environment**
  - Ease deployment and use on external systems
  - Ease the addition of new capabilities

- **Pave the way for further modernization efforts**
  - Workshop on code modernization for physical NWP models planned for 2026

ECMWF FORGE

Thank you! Any questions?

michael.lange@ecmwf.int
MLange805

ECMWF FORGE