System side meeting

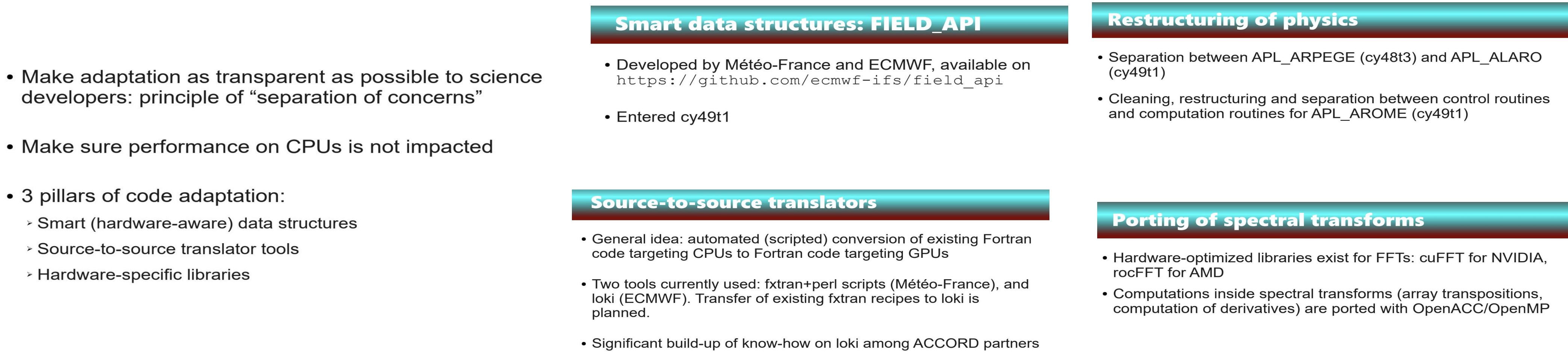
Facilitate the research activities: OOPS framework

- Impact: differences between MASTERODB (CNT0) and OOPS DA
- Export CY48T3_bf: AROME 3DVar include docs on how to build and run OOPS executable
- MENETRIER Benjamin: Making OOPS easier to use for new systems
- Pau Escribà OOPS Local Area 4DVAR
- Transfer of knowledge

System side meeting

Facilitate the research activities: GPU adaptations

- DEGRAUWE Daan: Progress in code adaptation to GPUs
- EL KHATIB Ryad: Météo-France RAPS2024 benchmark
- Transfer of knowledge



- Make adaptation as transparent as possible to science developers: principle of “separation of concerns”
- Make sure performance on CPUs is not impacted
- 3 pillars of code adaptation:
  - Smart (hardware-aware) data structures
  - Source-to-source translator tools
  - Hardware-specific libraries

Smart data structures: FIELD_API

- Developed by Météo-France and ECMWF, available on `https://github.com/ecmwf-ifs/field_api`
- Entered cy49t1

Source-to-source translators

- General idea: automated (scripted) conversion of existing Fortran code targeting CPUs to Fortran code targeting GPUs
- Two tools currently used: fxtran+perl scripts (Météo-France), and loki (ECMWF). Transfer of existing fxtran recipes to loki is planned.
- Significant build-up of know-how on loki among ACCORD partners

Restructuring of physics

- Separation between APL_ARPEGE (cy48t3) and APL_ALARO (cy49t1)
- Cleaning, restructuring and separation between control routines and computation routines for APL_AROME (cy49t1)

Porting of spectral transforms

- Hardware-optimized libraries exist for FFTs: cuFFT for NVIDIA, rocFFT for AMD
- Computations inside spectral transforms (array transpositions, computation of derivatives) are ported with OpenACC/OpenMP

System side meeting

Facilitate the research activities: GPU adaptations

- Fypp is a Python powered preprocessor. It can be used for any programming languages but its primary aim is to offer a Fortran preprocessor, which helps to extend Fortran with condititional compiling and template metaprogramming capabilities.

- Field API aims to ease the management and the transfer of data between CPUs and GPUs for the Météo-France/ECMWF software. The API is using fypp heavily to generate the code for several types and dimensions. It might look complicated, but if you are just using the API then you should not worry about it.

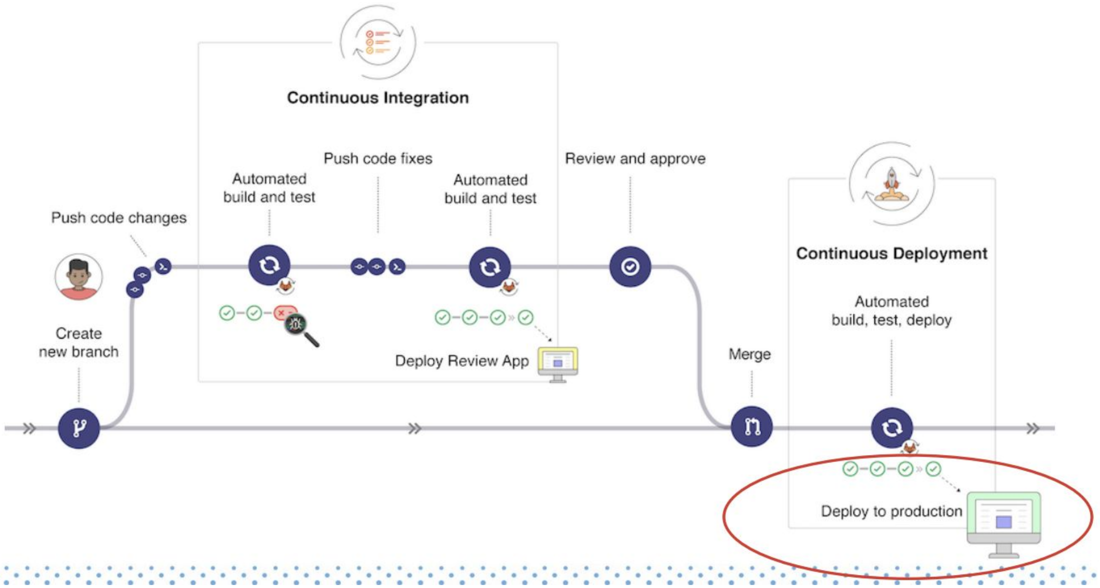
System side meeting

Facilitate the research activities: common system

- Ensures an **homogeneous and reproducible environmen**t for research activities and code debugging
- Ensures the **compatibility of a full stack of software** per version
- Ensures a minimum **tested workflow**
- Allows a **local deployment** to ensure the local installation is correct and could be use for research and operations
- Facilitates the **code maintenance** and increases the **code quality** (easier to test)
- Facilitates the **support and troubleshooting**
- Accelerates the research and development
- Increases the **sense of community** and helps to **homogenize the different levels** due to different resources

System side meeting

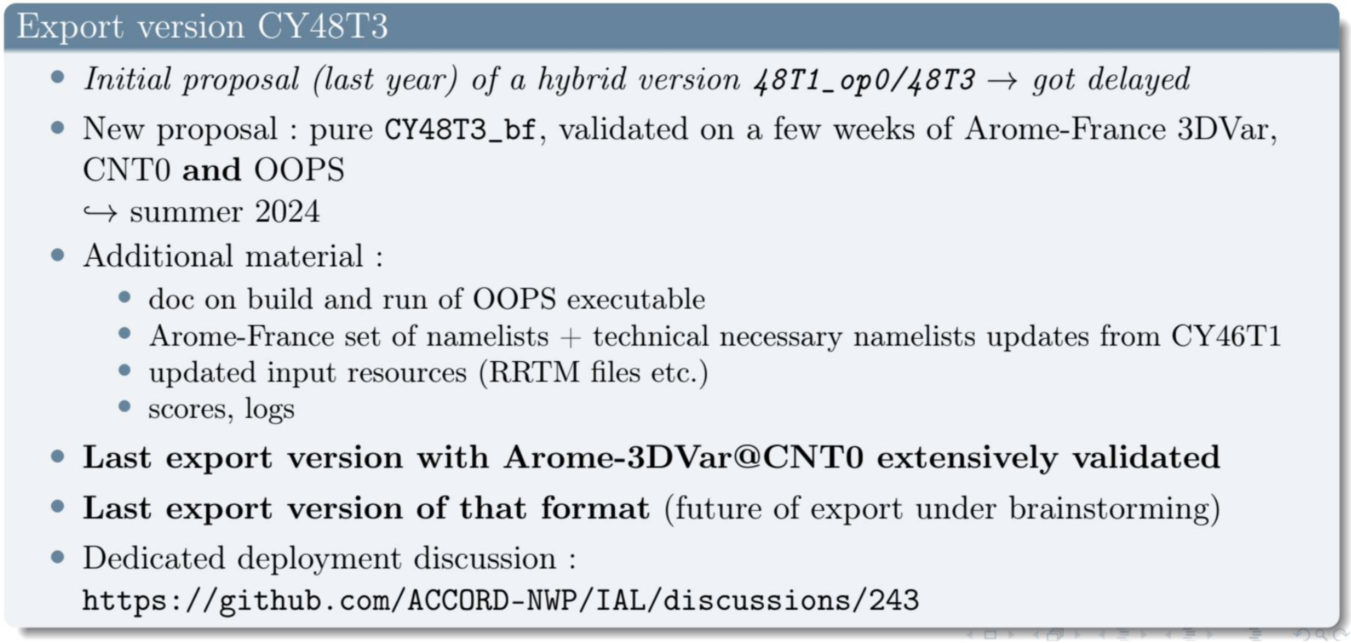
Facilitate the research activities: common system



- Synergies …
- HIRLAM make use of GitHub Actions
  - “*Automate your workflow from idea to production*”
  - Pull-requests compiled using CMake on Ubuntu 20 to help validate
- Could ACCORD pull requests make use of such Actions?
  - norms checker
  - build
  - test (Davaï and/or others)

System side meeting

Facilitate the research activities: release validation



Export version CY48T3

- *Initial proposal (last year) of a hybrid version 48T1_op0/48T3 → got delayed*
- New proposal : pure `CY48T3_bf`, validated on a few weeks of Arome-France 3DVar, CNT0 **and** OOPS
  ↪ summer 2024
- Additional material :
  - doc on build and run of OOPS executable
  - Arome-France set of namelists + technical necessary namelists updates from CY46T1
  - updated input resources (RRTM files etc.)
  - scores, logs
- **Last export version with Arome-3DVar@CNT0 extensively validated**
- **Last export version of that format** (future of export under brainstorming)
- Dedicated deployment discussion :
  `https://github.com/ACCORD-NWP/IAL/discussions/243`

Facilitate the research activities: release validation

Cycle

Source codes

94,2% Fortran codes

R cycle from Reading (ECMWF)

T cycle from Toulouse (MF/ACCORD)

- Code synchronization per cycle between repos in so called common cycle (name without letter)
- Technically validated by ECMWF and MF/ACCORD using Davaï
- Base code for contributions
- After declaration base code for export version (Aladin) op version (MF) and h-cycle(HIRLAM)

Export

Source codes and namelist

94,2% Fortran codes

Tx_export from MF

Includes the latest contributions from ACCORD

- Technically validated AROME version
- Meteorological quality assured over France with DA for a short season
- Base code for Operational implementation in Aladin countries

All CSC Export

Source codes and namelist AND tool (scripting) to run DA cycling

94,2% Fortran codes + Tool

Tx_export from ACCORD

Includes the latest contributions from ACCORD

- Technically validated AROME, ALARO and HARMONIE version
- Meteorological quality assured over France with DA for a short season for AROME, the same for ALARO and HARMONIE over LACE and HIRLAM domains
- Base code for Operational implementation and research for ACCORD

Tool for DA could be based on DEODE scripting

ALL CSC export should accelerate the use for research and development of the latest cycle

System side meeting

Facilitate the research activities

GIT :

- GitHub for ACCORD forge
- local support to implement GIT working practices

Git Forge webinar

DAP - Tech support visits for GIT

DAVAÏ:

- Dev working week
- Users training
- Training on ECMWF's HPCF for Davaï testers and integrators (spfracco project)

DAVAÏ training for users webinar

DAP- DAVAÏ contributors-developers WW

Vortex:

- Scientific visit

DAP - Visit to MF