RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

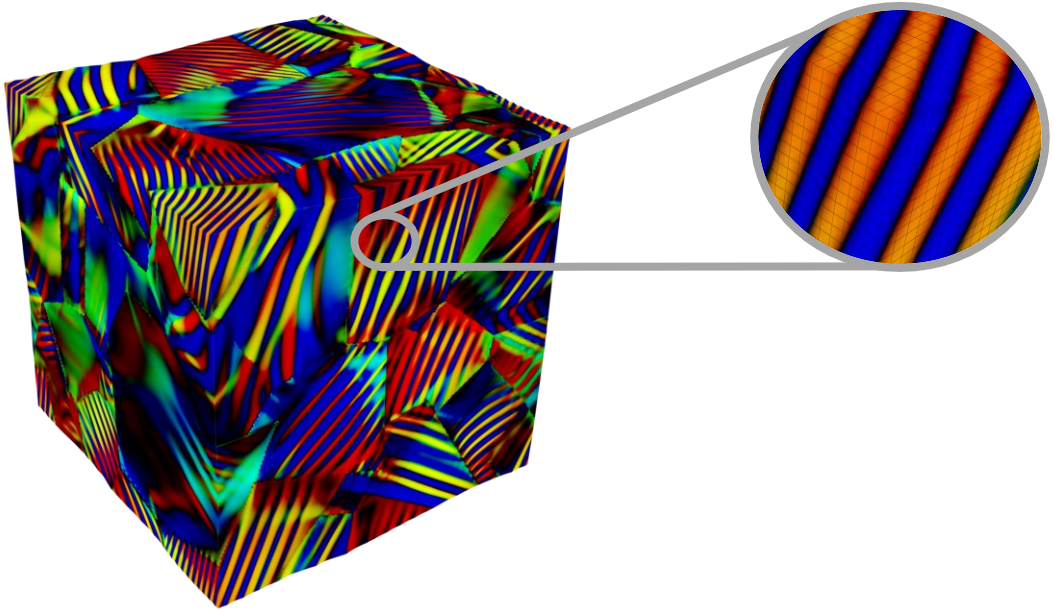
cea cnrs Inria

FRANCE 2030
PROGRAMME DE RECHERCHE
NUMÉRIQUE POUR L'EXASCALE

Restitution Brainstorming Coddex, Dyablo version in situ

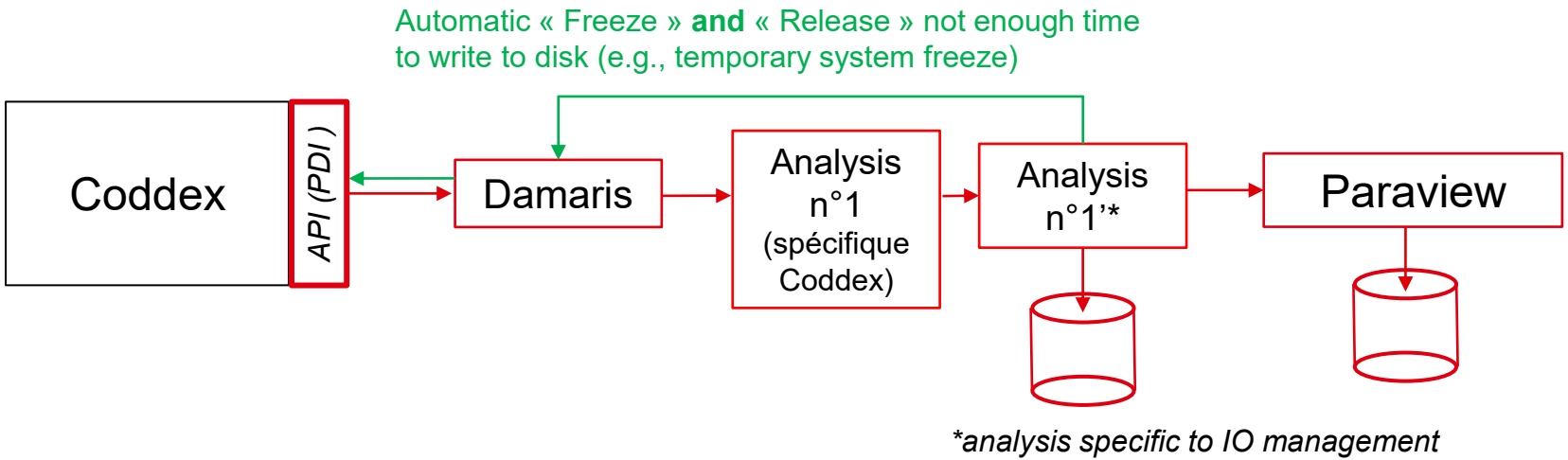
Laurent Colombet, Arnaud Durocher

Coddex



WP1 :

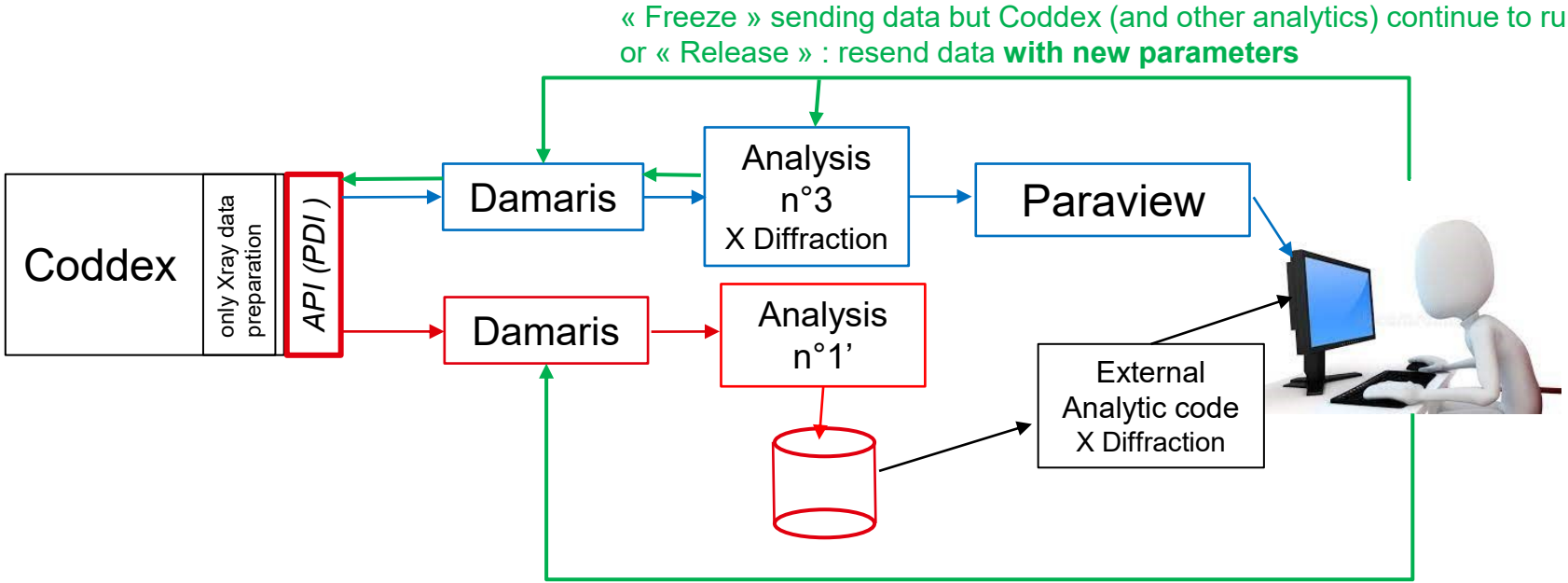
- **Develop an I/O analysis component (I/O orchestrator) to ensure data consistency and manage the in-memory data pipelining problem**
  - Anyway, the in-situ infrastructure has to be in place first
  - Unsure about who is going to do this
  - It may be related to the work with Gysela? We need to think more about this



**analysis specific to IO management*

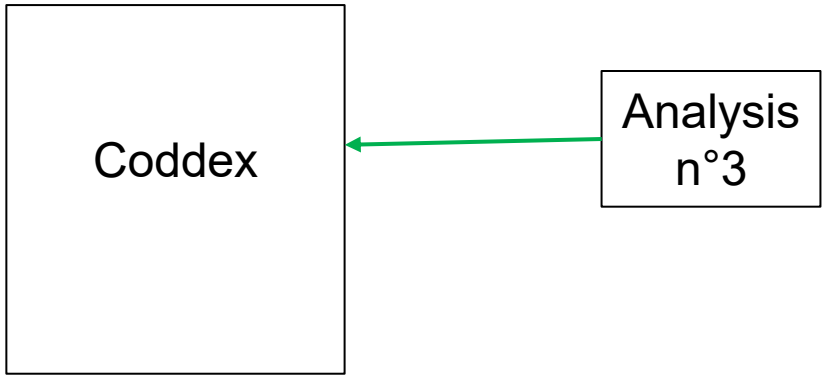
WP2 :

- When exploring results interactively in order to find the parameters for a method to be used in in situ analysis, is it possible to “freeze” the sending of data to this analysis?



WP2 :

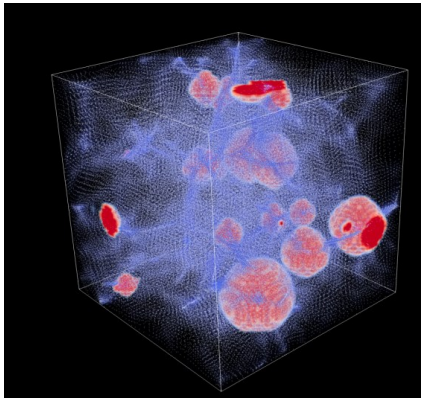
- From the Coddex code perspective, a priority for WP2 would be to develop the in situ scenario that allows an analysis to send feedback to the code.

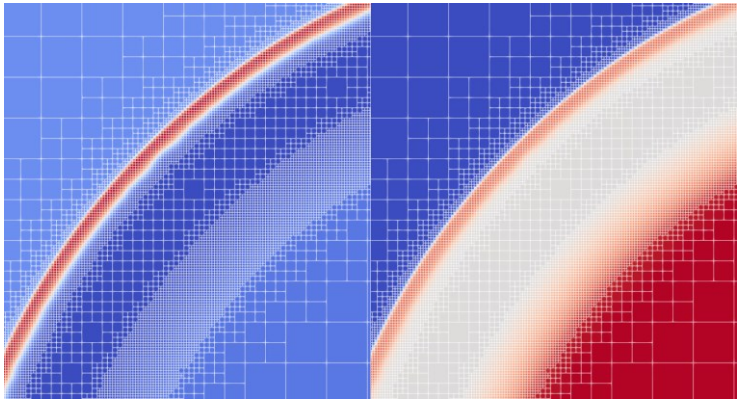


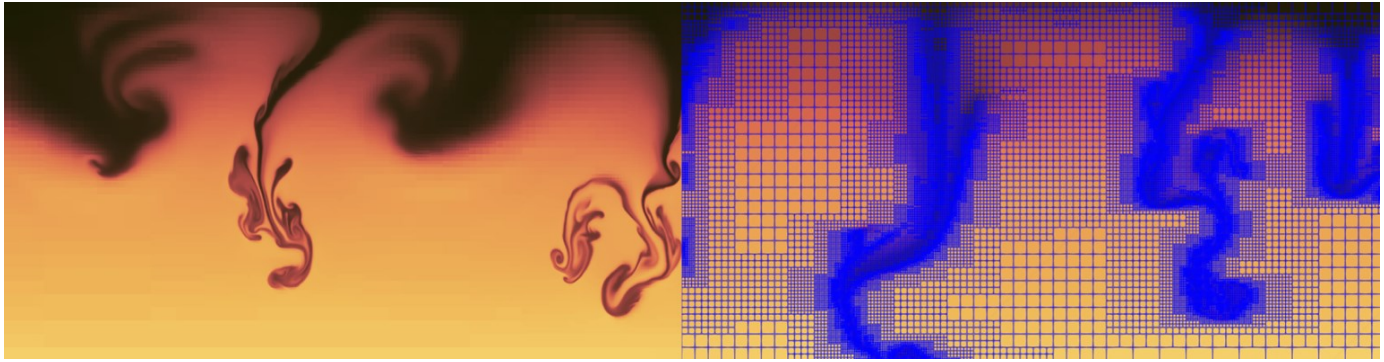
WP3 :

- **How AI can “help” Coddex analyses:**
  - **Event Detection :**
    - Detecting phase transitions in a material, such as the α - γ transition in cerium
  - **Anomaly detection**
    - Help to find “non-physical” situation and trigger a specific response from Coddex
  - **Simulation-based inference (P1)**:
    - Propose an initial relevant setting for the algorithm simulating X-ray diffraction
    - Learning from smaller scale simulations and generalize on larger ones

Dyablo







Develop new AMR data formats

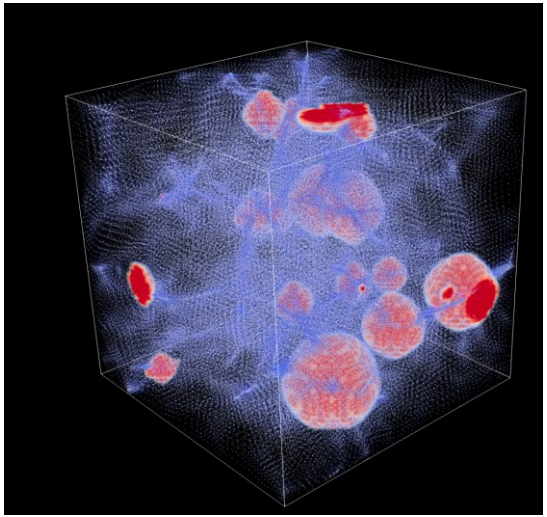
Post-Doc financed by Exa-DoST

Currently in Dyablo

- Post-processing outputs use ***Paraview Unstructured Mesh + HDF5***

→ Heavy geometry / connectivity (more disc space than actual data)
→ Slow post-processing
→ Large data loaded all at once through paraview

But makes nice images for small simulations



Cosmology : 4 Mpc box with dark matter particles and ionization, image generated by paraview

Wishlist for a new format dedicated to AMR

- Efficient storage of AMR grid (implicit geometry)
- Allow efficient and fast post-processing with AMR
  - → Leverage hierarchical structure (Level of detail, …)
  - → AMR-optimized algorithms (Slices, subdomains, …)
- Works for distributed simulations
  - → Independent of simulation MPI parallelism
  - → Load partial data (less RAM than full simulation)
- Standardization : integrate with existing tools

(few leads : Paraview HyperTreeGrid, yt, …)

Post-Doc : Sylvain JOUBE

- 18 month : nov. 2025 → 2027
- AMR ***data*** format
- Implementations : Dyablo (CEA), Samurai (Polytechnique)
- Associated post-processing tools (?)

→ **Exa-DoST expertise will be precious to help implementation**

General :

- **Collaborate through the post-doc (Financed through WP1)**
  - Co-design the data format with IO libraries / experts
  - Standardization/data format is specific to AMR : we probably have to do it on out own
  - Help him implement actual disk writes (maybe PDI)
- **We are open to provide a mini-app / benchmark with dyablo**
  - We can generate a lot of data (bench upt to 2000 GPUs on AdAstra)

In-situ / AI :

Dyablo is still young and post processing / AI analysis are not ready yet

- Use-cases (post-processings, …) will come later from physicists / dyablo applications
- We want to make sure everything we make for IO is ready for in-situ / AI

=> Pave the way to in-situ, AI, … by using Exa-Dost software