

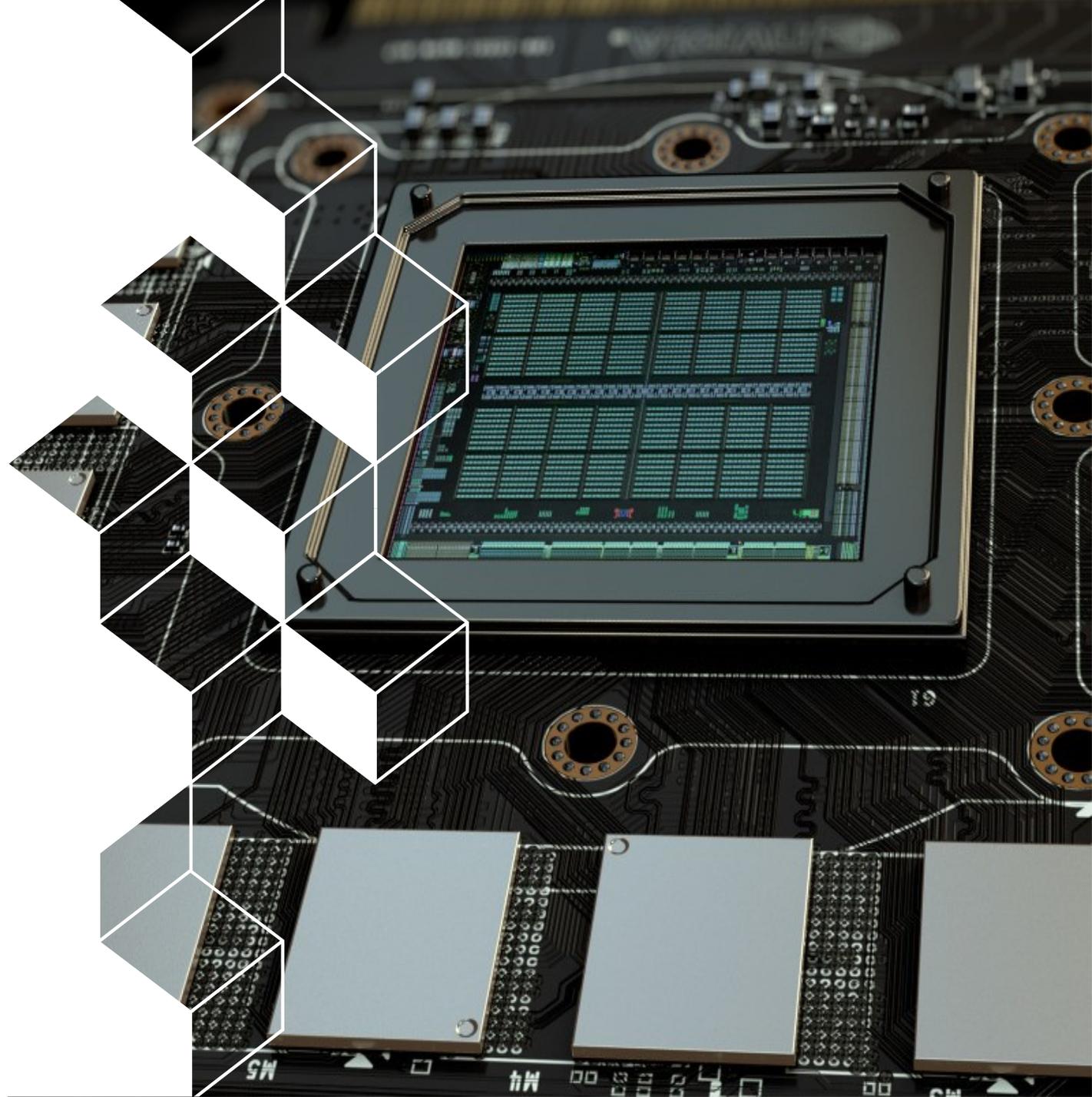


# Moonshot CExA, Calcul Exascale Au CEA

Catalyseur logiciel pour calcul GPU

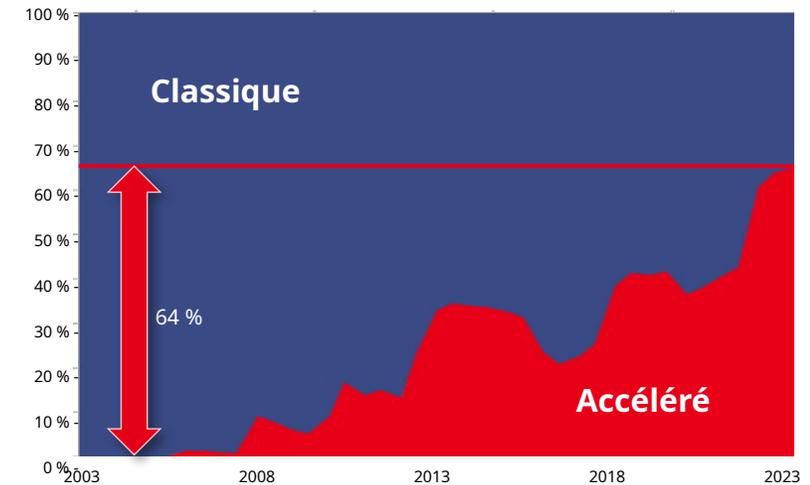


*Mission numérique - 6 octobre 2023*  
*Julien Bigot & l'équipe CExA*

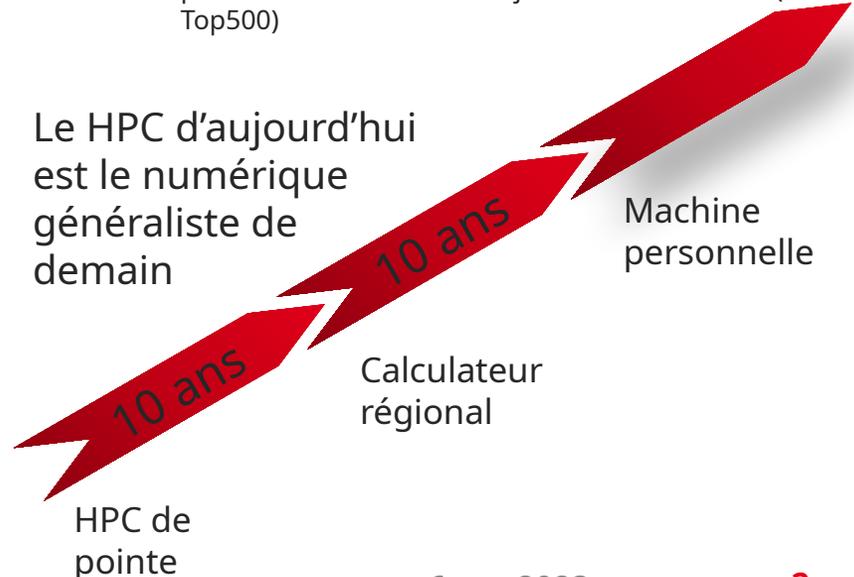


# Contexte

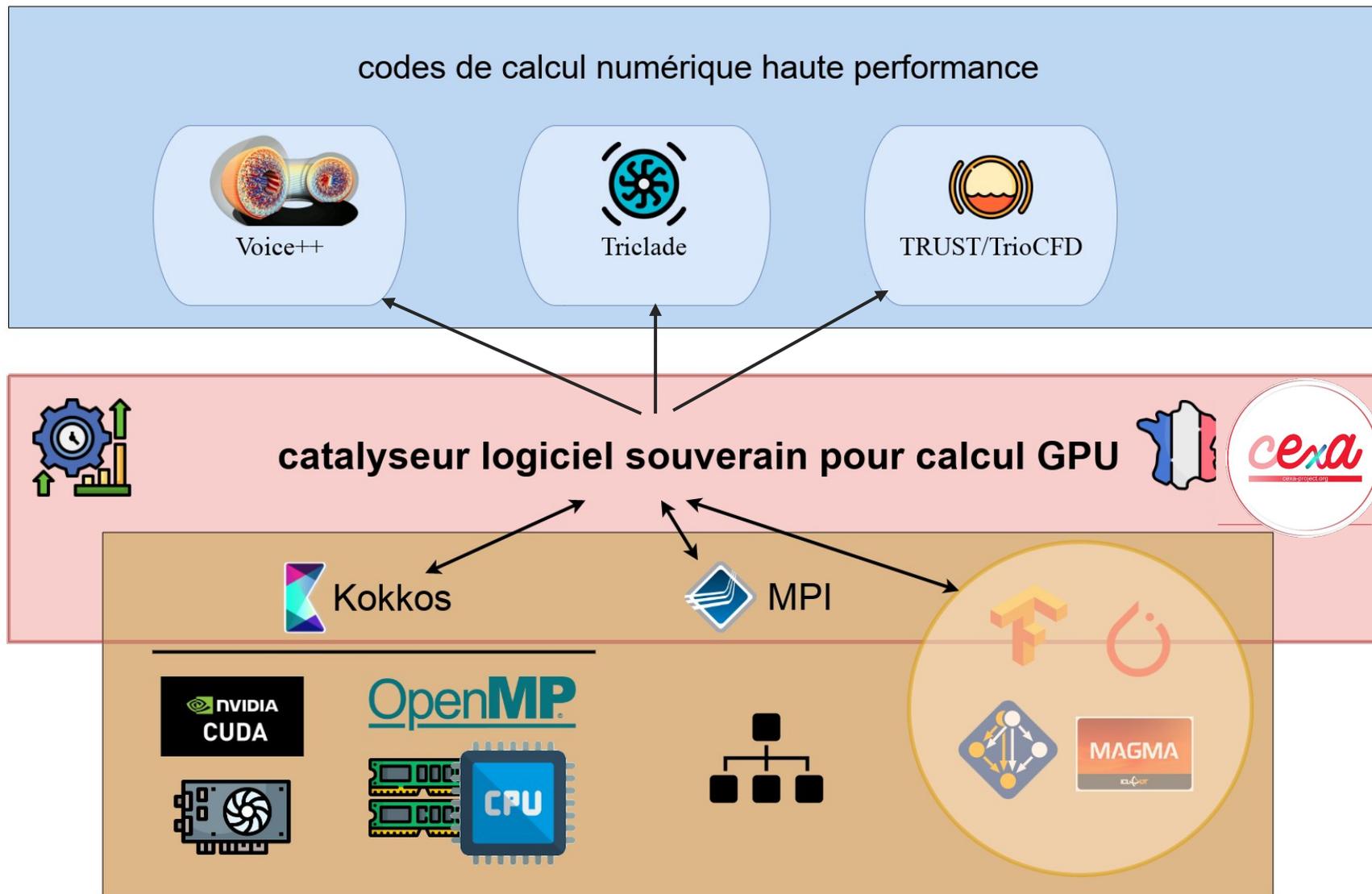
- Le calcul intensif : un outil transverse de **souveraineté** et de **compétitivité**
  - Jumeaux numériques, modélisation du climat, dissuasion nucléaire, physique aux échelles extrêmes, conception multi-échelle des matériaux, médecine personnalisée, respect de la vie privée, etc.
  - Largement dans la société, transverse à **toutes les DOs** du CEA
- Arrivée de l'**Exascale** (1ère machine cette année)
  - Architectures accélérées (GPU)
  - Arrivée des premiers supercalculateurs en Europe pour 2024-2025
    - Une machine Exascale en France au CEA/TGCC
  - Besoin de re-développer les applications pour en tirer parti
- Les intergiciels GPU : des **catalyseurs**
  - **Portabilité de performances**
  - Aux États-Unis : porté par l'*Exascale Computing Project* (ECP) ⇒ Kokkos
    - Stratégie open-source pour assurer un transfert vers les industriels
  - En Europe et en France : de la recherche, mais pas d'outil concret
- Un **besoin criant** de solution souveraine
  - **Maîtriser** la feuille de route, **adapter** à nos spécificités matérielles et logicielles



Puissance de calcul des 500 supercalculateurs les plus puissants au monde de juin 2003 à 2023 (source Top500)



# Le projet



Three stacked chevron-shaped boxes containing project goals:

- Top (Light Blue):** Diffuser et former à **CExA** au sein du CEA et plus largement
- Middle (Dark Blue):** Adapter démonstrateurs applicatifs intégrant **CExA**
- Bottom (Light Blue):** Construire un catalyseur logiciel souverain pour le calcul GPU : **CExA**

# CExA en bref

## Stratégie “adopter et adapter” basée sur Kokkos

- Kokkos : une **base solide**
  - Mature, libre et open-source
    - Une architecture de portabilité de performance
      - Prête pour les architectures futures
    - Une démarche d'intégration au standard C++
      - Un tremplin vers le standard C++
      - Une preview en avance de phase du C++ parallèle
- Des **adaptations nécessaires**
  - Pour le **matériel** Européen
    - Pas de souveraineté matérielle sans logiciel souverain
  - Pour les **applications** Européennes et CEA
    - Prendre en compte nos spécificités

### Adéquation aux architectures « Mémoire distribuée »

- Portabilité de performance MPI+Kokkos CPU & GPU
- Transferts mémoires efficaces
- Supports virtualisation GPU

### Interface avec les outils externes de traitement de données

- Interface avec Pytorch, Tensorflow
- Outils d'algèbre linéaire par lot

### Support des architectures hétérogènes

- Support de code multi-architectures
- Support multi-espaces d'exécution
- Support des processeurs EPI

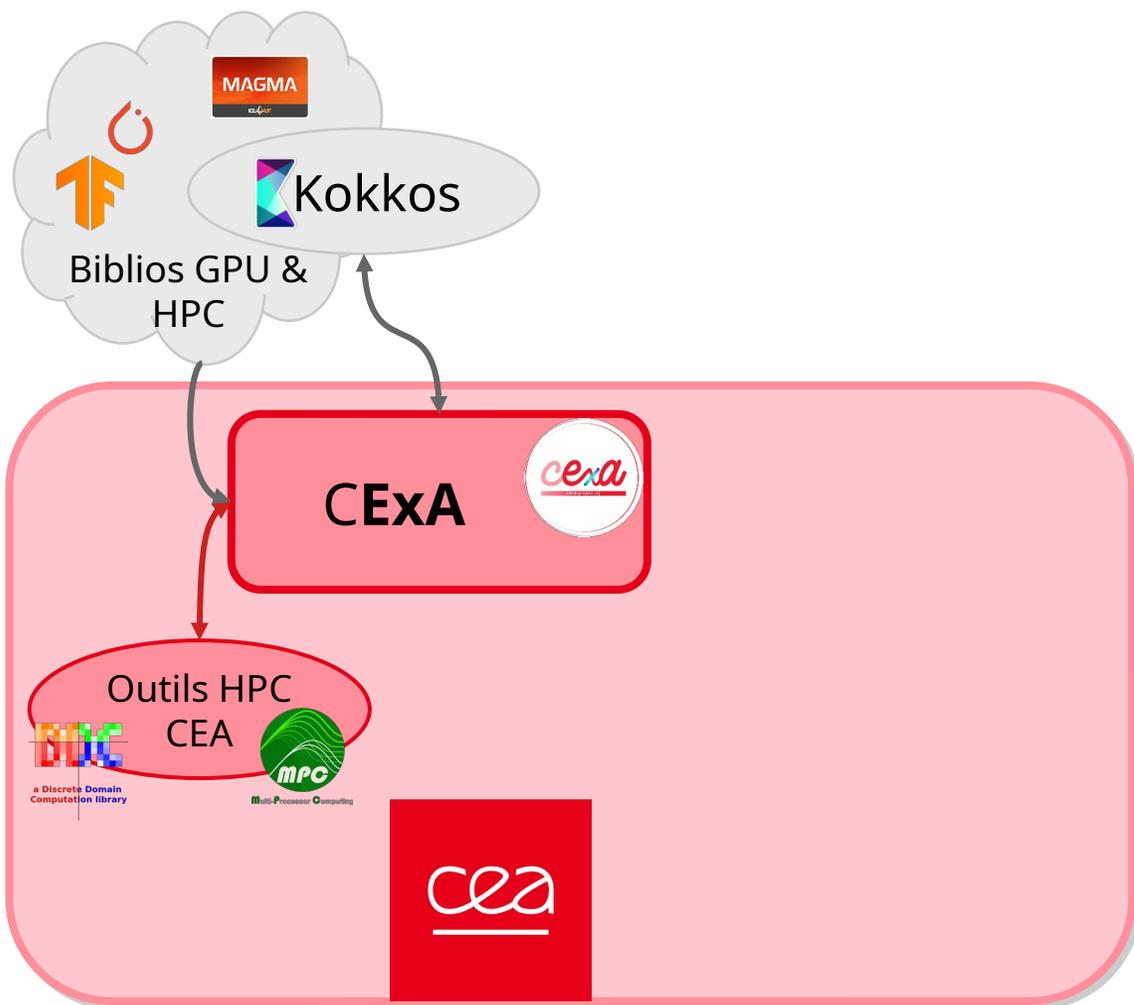
### Simplification du déploiement sur nos machines

- Gestion du déploiement multi-device
- Intégration continue et installation sur nos calculateurs

Spécificités matérielles

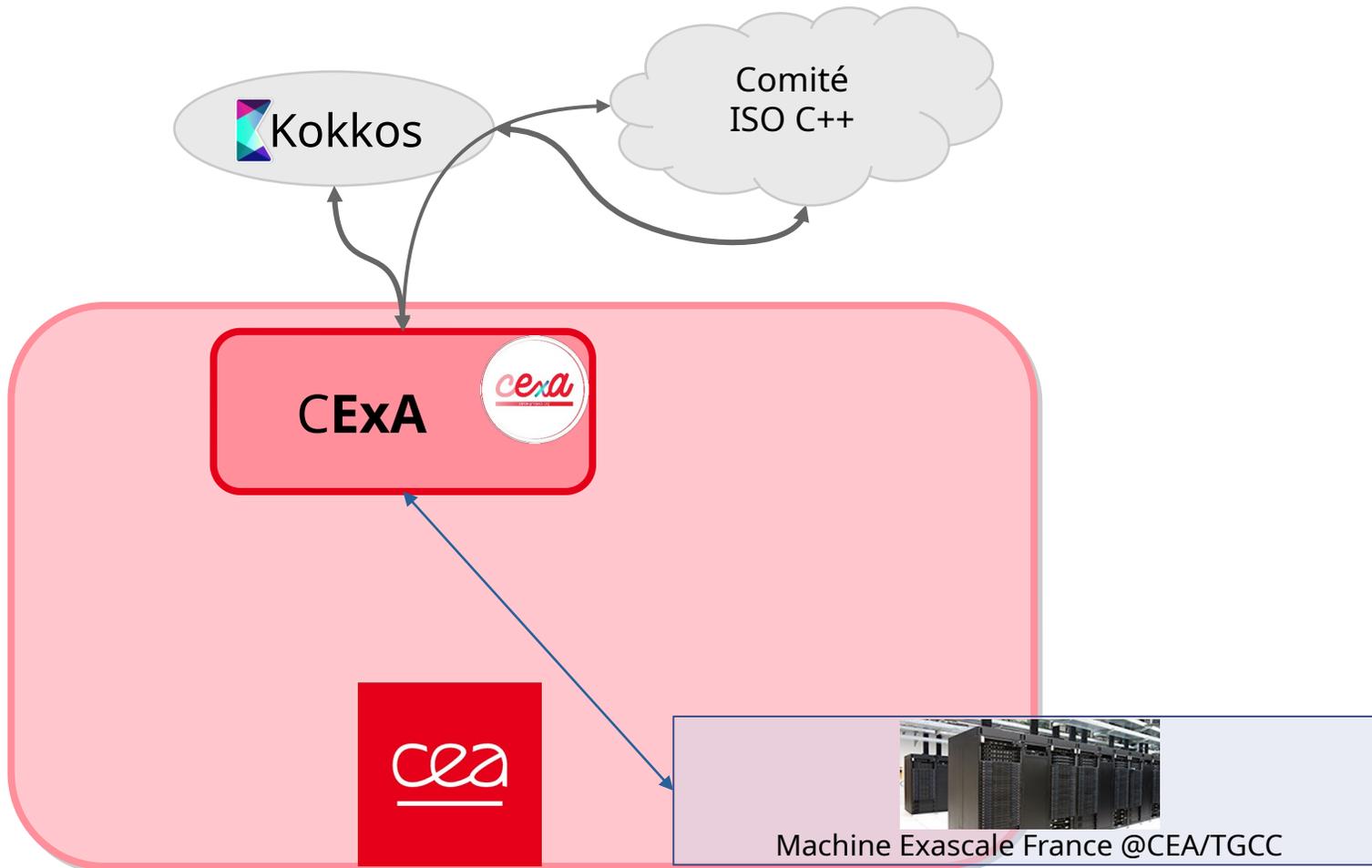
Spécificités logicielles

# L'Écosystème de CExA : l'Amont



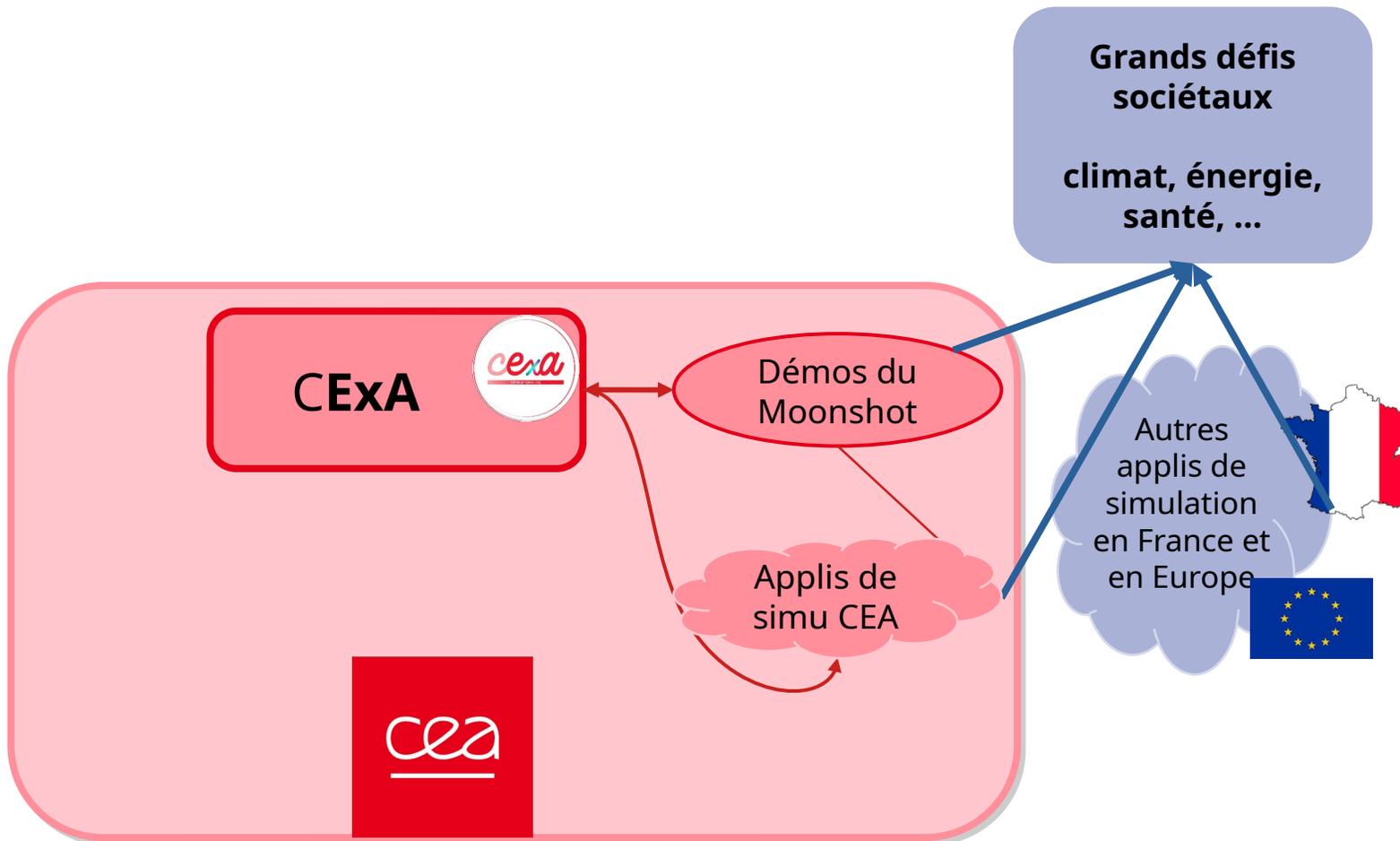
- Bibliothèques GPU & HPC
  - Tensorflow, Pytorch, MAGMA, etc.
  - Interfaçage grâce à l'aspect libre & open-source
- Équipe de développement Kokkos
  - Lien fort établi
  - Présent aujourd'hui
- Bibliothèques HPC CEA
  - MPC, DDC, Arcane, etc.
  - Intégration et échanges

# L'Écosystème de CExA : les partenaires



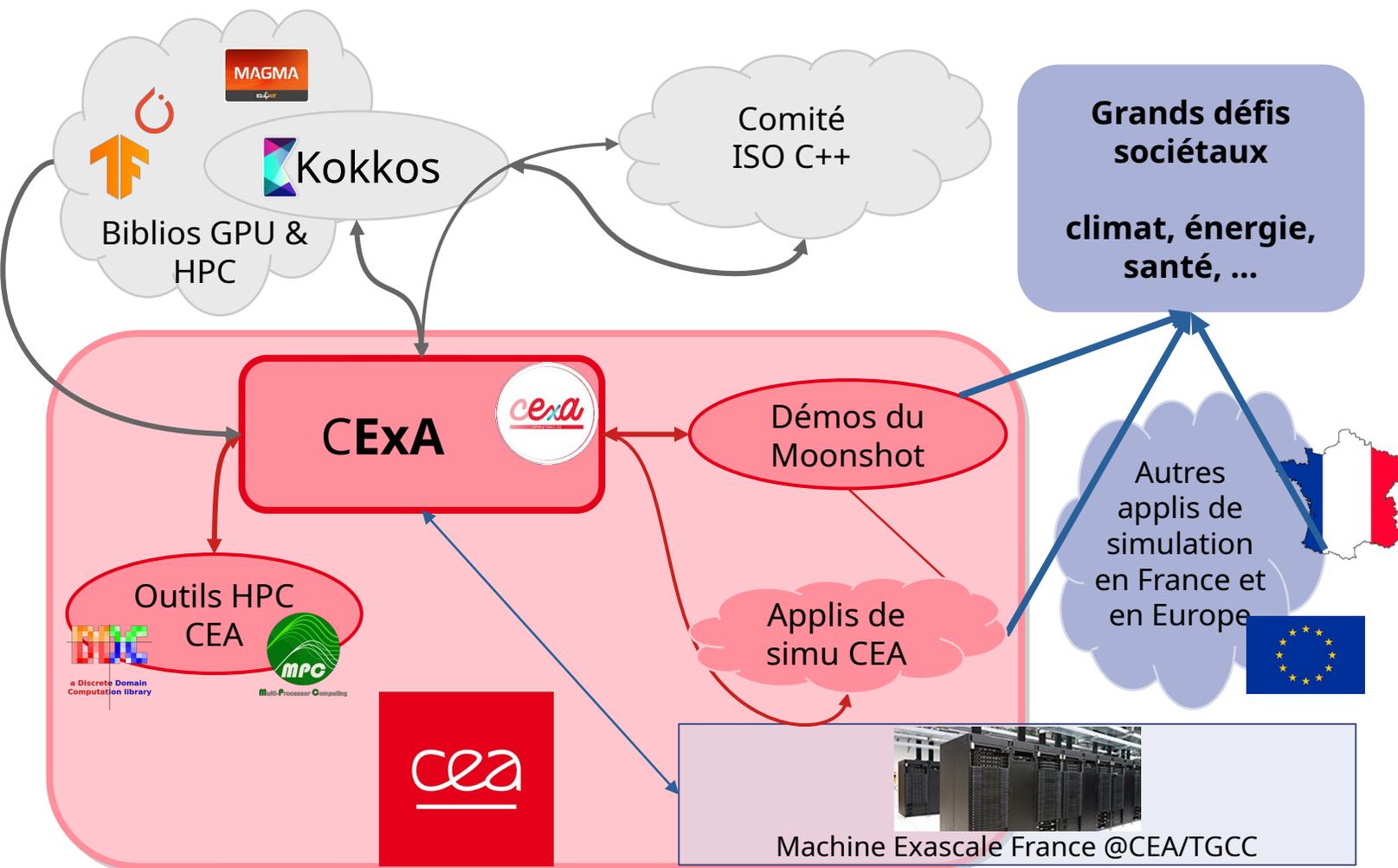
- Kokkos & comité ISO C++
  - Standardisation
    - au travers Kokkos
  - Normalisation et pérennisation des approches CEA
- Projet Jules Vernes (Exa France)
  - Liens forts avec GENCI, le TGCC et NumPEX
  - AAP fin 2023
    - exigence de CExA
  - Réponse 2024
    - Choix de l'architecture prioritaire
  - Livraison fin 2025
    - CExA prêt à la production

# L'Écosystème de CExA : l'Aval



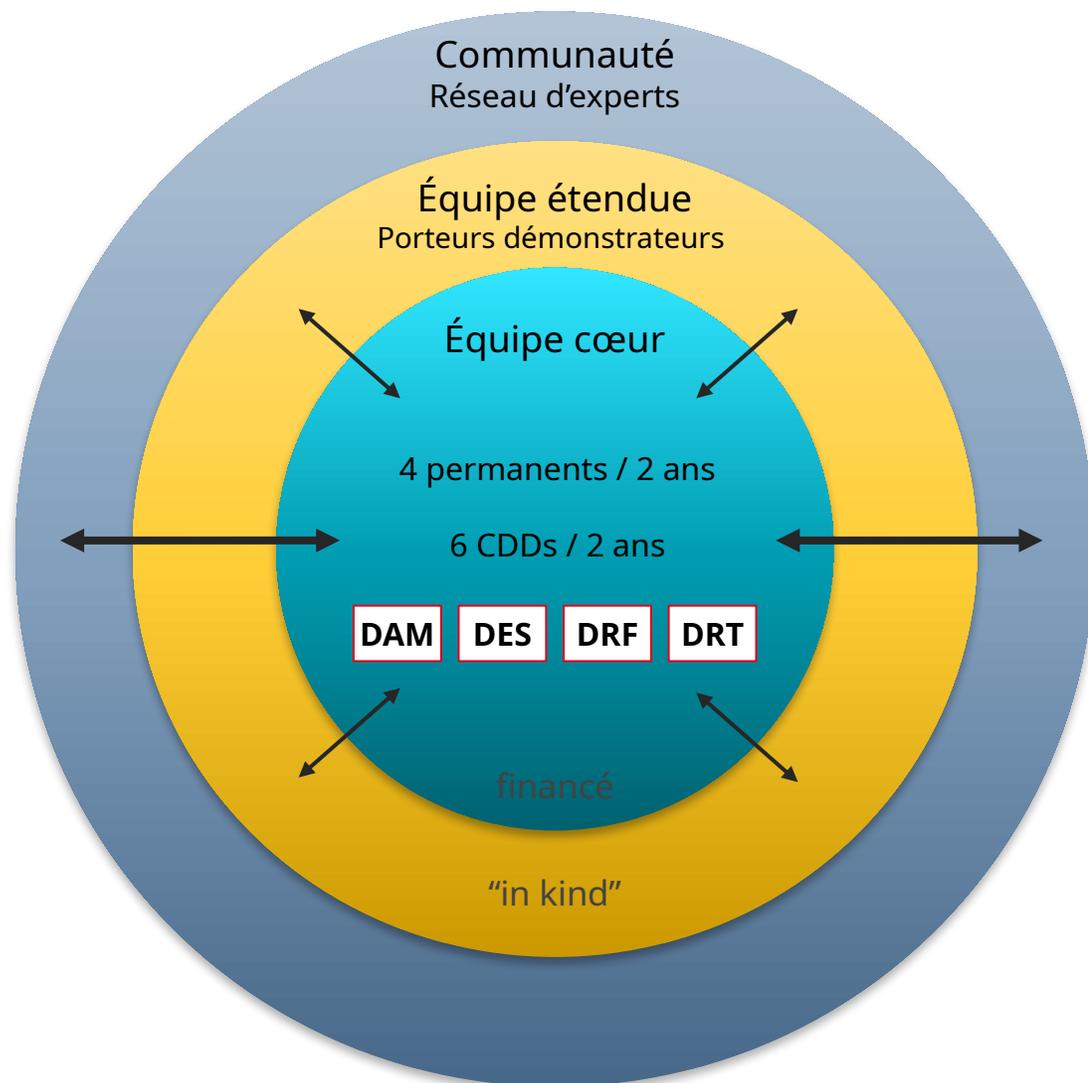
- Aval à deux niveaux
  - Étage d'accélération ⇒ applications
  - Second étage ⇒ enjeux sociétaux
- Démonstrateurs intégrés
  - Co-développement en binôme
  - Formation des équipes
  - Retombées dans domaines prioritaires
- Applications CEA
  - Formation, hackathons, expérience
  - Effet d'entraînement, choix clair
  - Création d'une communauté
  - Contribution CEA ⇒ défis sociétaux
- Communauté FR et EU
  - Visibilité et place du CEA

# L'Écosystème de CExA : l'Après



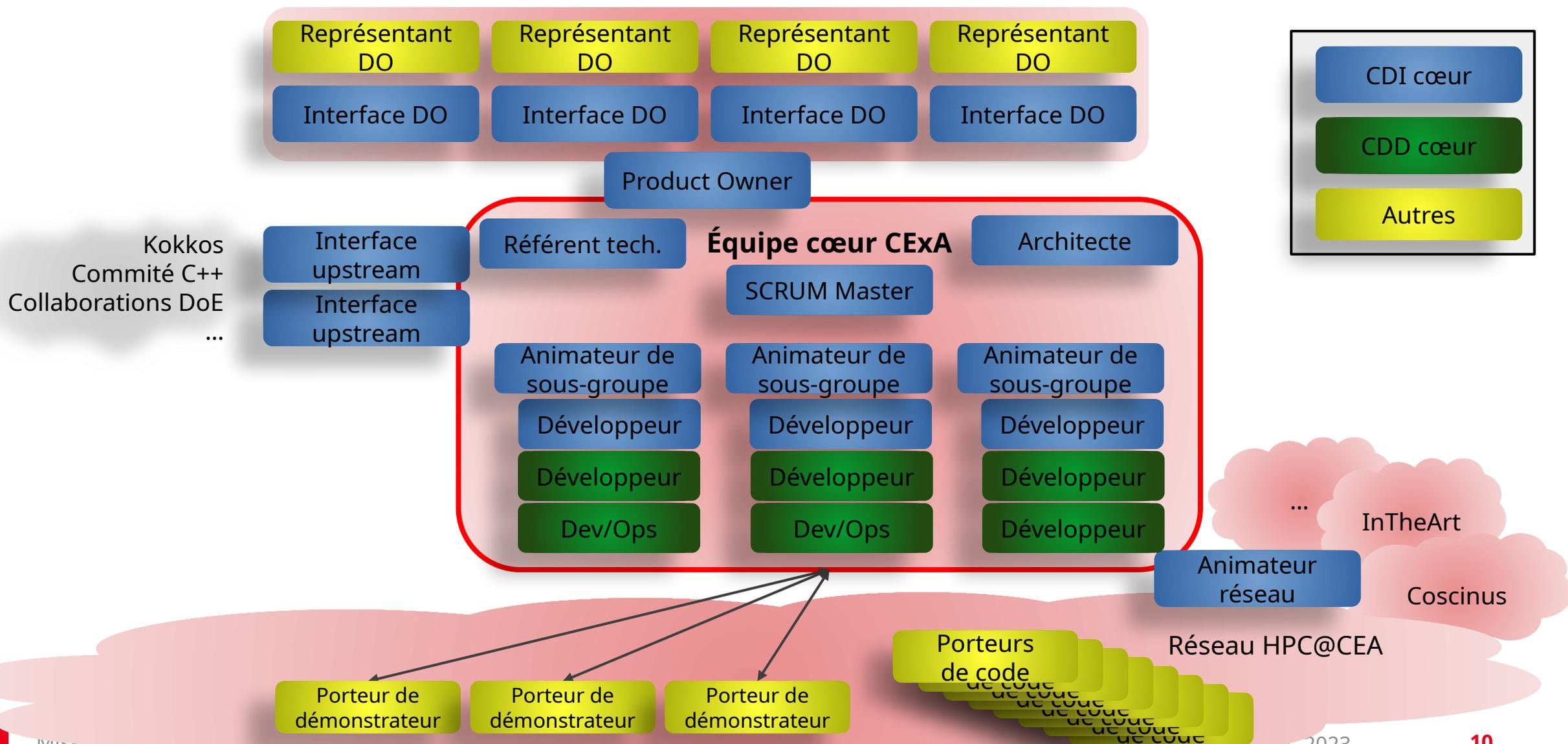
- Préparation de l'outil pour le calcul numérique sur GPU
  - Après les graphismes (milieu 1990)
  - Après les réseaux de neurones (fin 2000)
- Positionnement au cœur de la pile logicielle
  - Expertise de pointe sur l'outil
  - En avance de phase
  - Avec une feuille de route adaptée
- Un avantage compétitif unique pour les décennies à venir

# L'organisation du projet



- **Équipe cœur**
  - Pilotage, Réalisation et Dissémination
  - 4 ETPs permanents financés pendant 2 ans
  - 6 ETPs CDD financés pendant 2 ans
  - Croissance organique de l'équipe
- **Équipe étendue**
  - Intègre les porteurs de démonstrateurs
  - 1 par DO (hors budget / ressource "in kind")
  - ~6 mois
- **Communauté**
  - Fédération d'un réseau d'experts
  - Co-conception de CExA:
    - Remontée des besoins
    - Redescente de CExA dans les applications
  - Cible privilégiée de la dissémination
  - Pérennisation des travaux

# Notre organisation agile



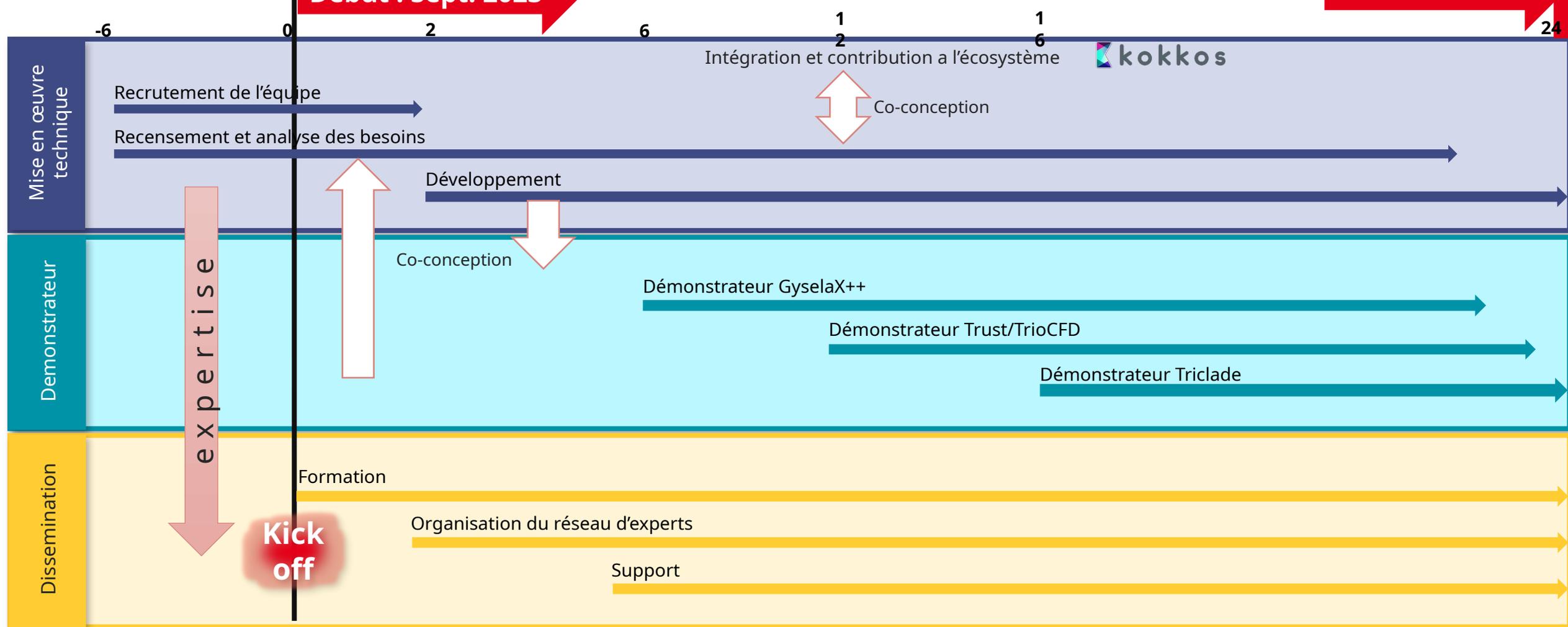
# Planning

Préparation préalable

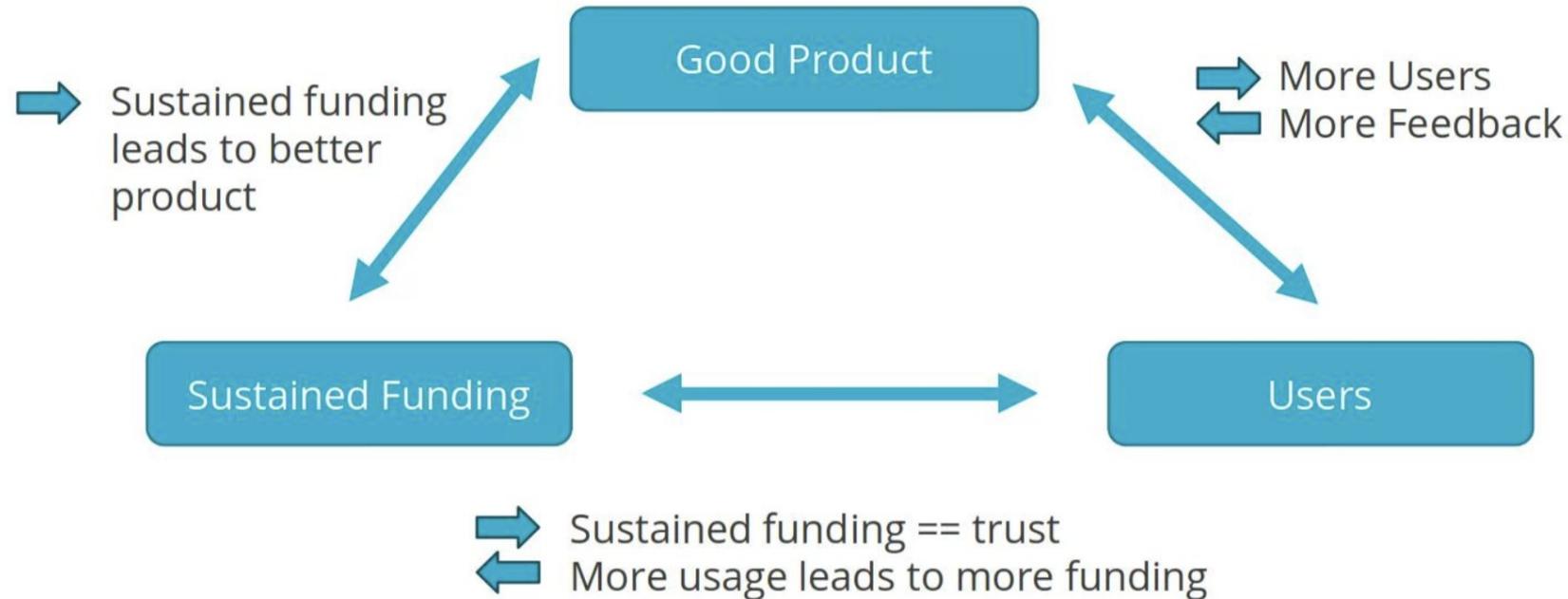
Début : Sept. 2023

Durée : 2 ans

Fin: Août 2025



# Sustainment a virtuous self-reinforcing cycle

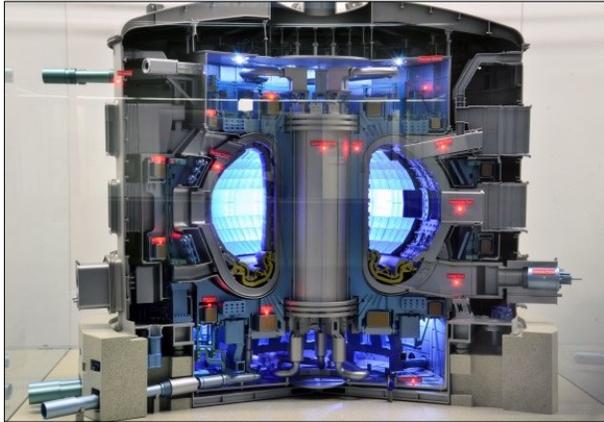


**There is strength in numbers: collaboration on core product good for everyone!**

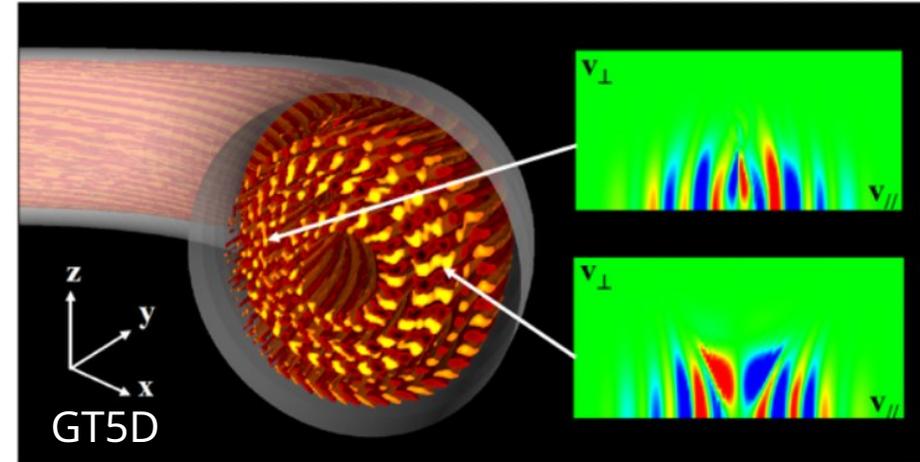
©  
Christian Trott (LLNL) &  
Damien Lebrun-Grandie (ORNL)  
US DoE

# GyselaX++ : Démonstrateur #1 (DRF)

Projet ITER



Gyrokinetic plasma turbulence simulations



Pour optimiser les performances et minimiser les risques, chaque scenario ITER devra être et sera valide numériquement

Une chaîne complète d'outils numériques sera requise,

- Allant des modèles d'échelle, utilisable en temps réel
- Aux simulations premier principes, coûteuses mais fiables

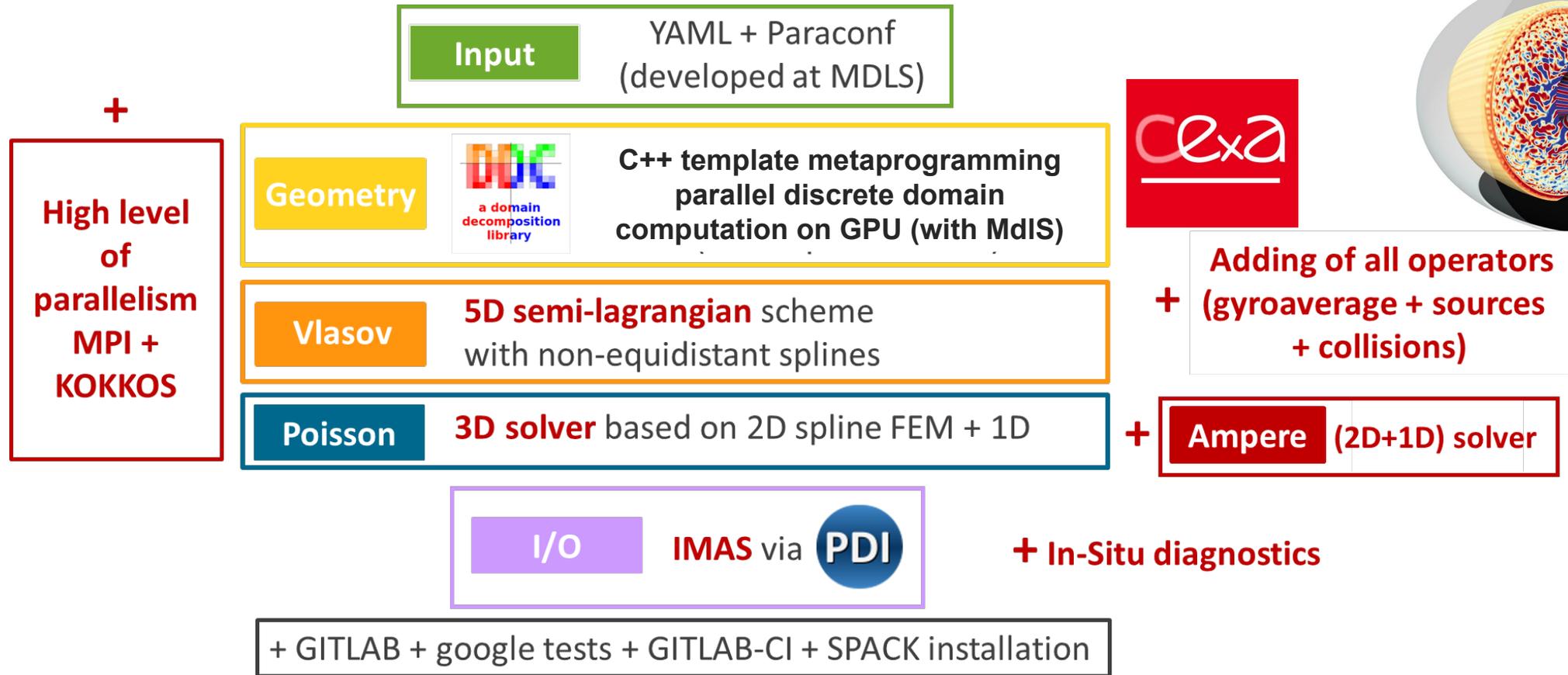
Le transport turbulent gouverne le confinement mais les plasmas de Tokamak sont faiblement collisionnels

- L'approche cinétique est requise

Gysela est le seul code semi-Lagrangien au monde qui permette ces simulations

# GyselaX++ : Démonstrateur #1 (DRF)

Code 5D en C++ modern, scalable sur architectures Exascales



# TRUST/TrioCFD : Démonstrateur #2 (DES)

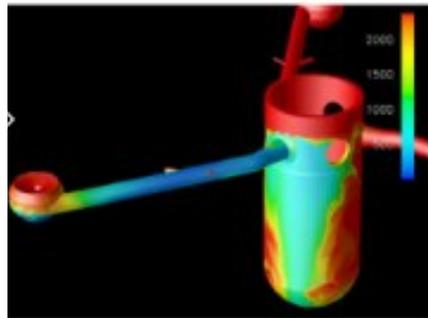
**TRUST** Plateforme de thermohydraulique (DES/DM2S/SGLS/LCAN)

**Trio** Application TRUST dédiée à la CFD

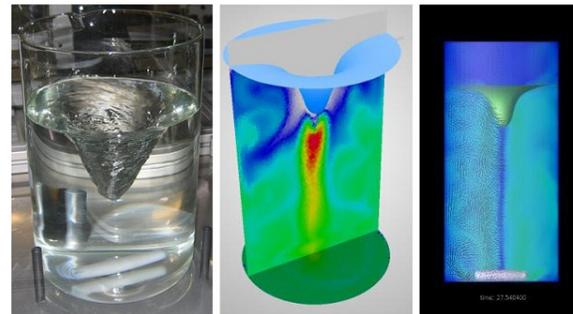
Mécanique des fluides

- Incompressibles/faiblement compressibles
- Mono/Diphasiques
- Suivi d'interfaces

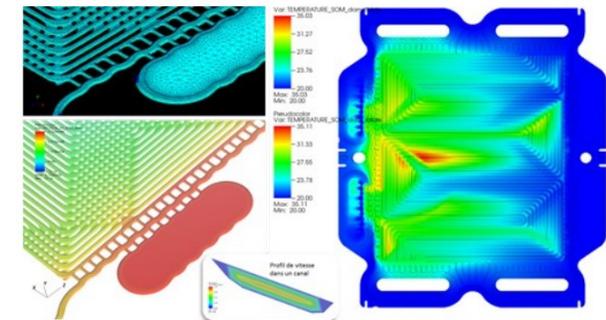
Domaines d'application



Réacteur REP



Agitateur à effet vortex



Pile à combustible

C++, MPI, OpenSource (Github)

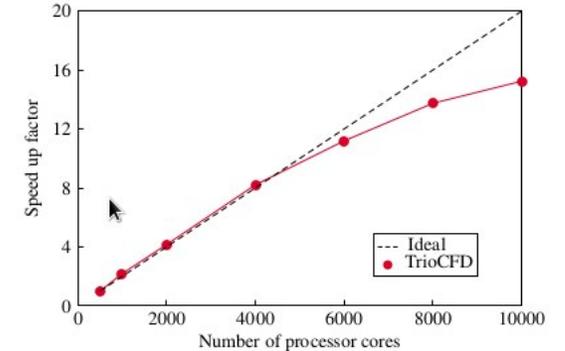
# TRUST/TrioCFD : Démonstrateur #2 (DES)

## Capacités actuelles HPC:

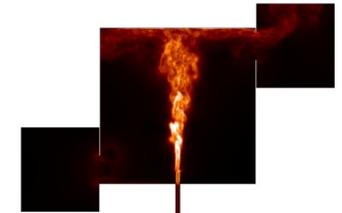
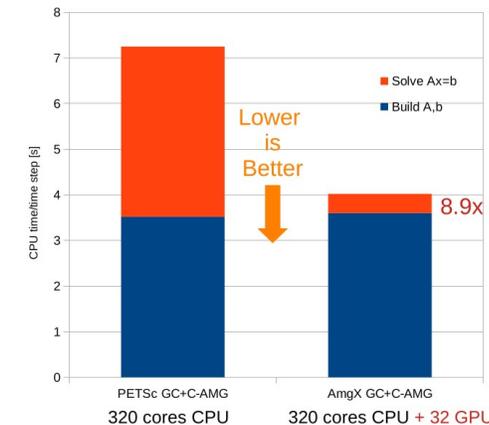
- MPI:
  - 2e9 DDL, 50K coeurs (Irene Jolliot–AMD Rome)
- MPI/GPU:
  - 80 107 DDL, 320 coeurs–32 GPUs (Irene Jolliot–V100)
  - Via librairies algèbre linéaire (Nvidia AmgX, AMD rocALUTION)
  - Directives OpenMP en cours de test

## Apports de CExA:

- Ecriture plus élégante/performance des noyaux OpenMP
- Parallélisme hybride MPI/OpenMP
- Vectorisation SIMD
- Portabilité de la performance (GPU Nvidia, AMD, Intel)
- Déploiement simplifié sur calculateurs CEA



DNS simulation (TrioCFD 1.8.3) on Irene Jolliot cluster (TGCC)



- ▶ Mini-GAMELAN geometry
- ▶ Structured mesh (VDF)
- ▶ 80M cells (250K/core)
- ▶ Unsteady DNS
- ▶ GC + C-AMG solver
- ▶ 50% time into solver

# Triclade : Demonstrateur #3 (DAM)

Code C++ pour la résolution des équations Navier-Stokes compressibles avec 2 constituants

Maillage 3D cartésien structuré

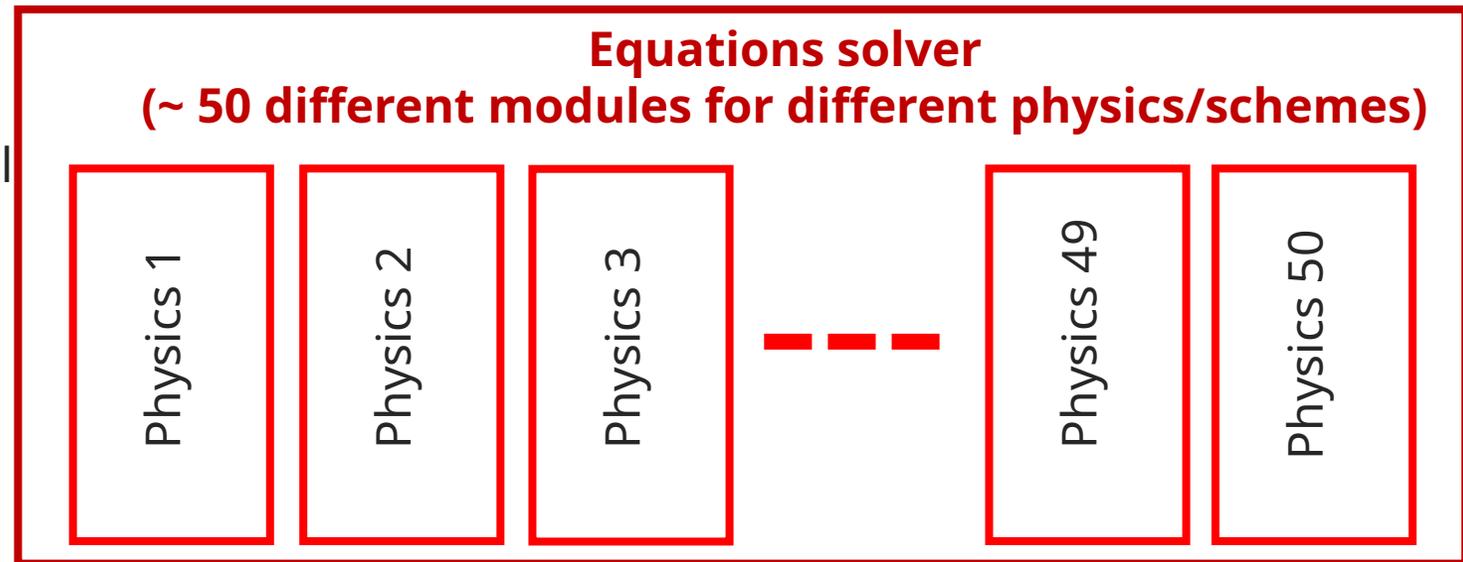
- 10243 mailles
- ~ 8000 processus MPI

3 parties

- 100K lignes de codes au total pour la partie centrale du code

Focus sur un seul module

- Le seul utilisé dans les cas représentatifs visés
- ~ 10K lignes de codes



C++  
MPI



MPI I/O

# Triclade : Démonstrateur #3 (DAM)

Il n'existe pas encore de support GPU dans Triclade

Triclade fournit des propriétés intéressantes pour le parallélisme en mémoire partagée / GPU

- Le module cible n'utilise pas bibliothèques externes
  - Pas de problème d'intégration lors du portage vers des noyaux GPUs
- Le code fait des calculs sur les 3 directions
  - Les calculs sur les 3 directions semblent indépendants
  - Ils sont donc probablement parallélisable efficacement sur GPU

Apports de CExA:

- Portage du module utilisé en Kokkos pour fournir une version GPU
- La version Kokkos peut apporter une version mémoire partagée/thread si aucun GPU n'est disponible
- Vectorisation SIMD
- La version Kokkos permet une portabilité du code sur différents types de GPUs

# A first draft of EPICs (1)

Introduce physical variables management to write more robust simulation applications

- Introduce properties and operate directly on such views
- Kokkos view manager (ease to develop, readability, Maintainability)
- Perform batching on physical variables
- Filter on properties

Facilitate port legacy of applications to accelerators hardware (GPU)

- Commonly used libraries/frameworks (e.g. Linear Algebra) (bridge CPU modules to operate on Kokkos structures)
- Automatic memory copy to GPU when required
- Diagnostic management (Code Profiling)

Offer support to advanced and state of the art 3rd party functions/libraries (each vendor has its own library, plug to the right library via Kokkos level interfaces/adapters)

- Integrate FFT (via CExA adapter / support Kokkos compatible API)
- Integrate Spline (redevelop using Kokkos, and integrate)
- Connect to AI libraries (e.g. PyTorch)
- Solve Linear Algebra problems with Kokkos

Make full use of current and future European Exascale architectures

- Adapt to unique memory architecture
- Improve interoperability and performance between Kokkos and distributed parallelism (e.g. GPU direct, Remote Space, MPI, etc.)
- Improve performance and execution on ARM based technologies for HPC (Grace ARM cpu (SVE vectorization), RHEA ARM cpu, A64FX cpu)
- Improve performance on x86 cpu (vectorization)

# A first draft of EPICs (2)

Extend programming model to cover more usage scenarios

- Multi-device management (abstract multi GPU) In one node: 1 CPU process can send information to all GPUs of the node
- (Heterogeneous hardware, e.g. AI specific GPUs / NPUs)

Improve scientific applications Development by introducing Continuous Integration Facility

- CI / CD facilities installation
- Methodology
- GitOps implementation

Use Cases improvements (KPIs)

- Performance Improvements / Ports
- Readability / Maintainability
- Tooling (code profiling)
- Robustness (Unit tests)

Support CEA Technical Community

- Community (Web site with all libraries that exist)
- Tests / Investigations ? State of the art (Bibliography, experimental libraries, features that come with Kokkos)

# Pour conclure



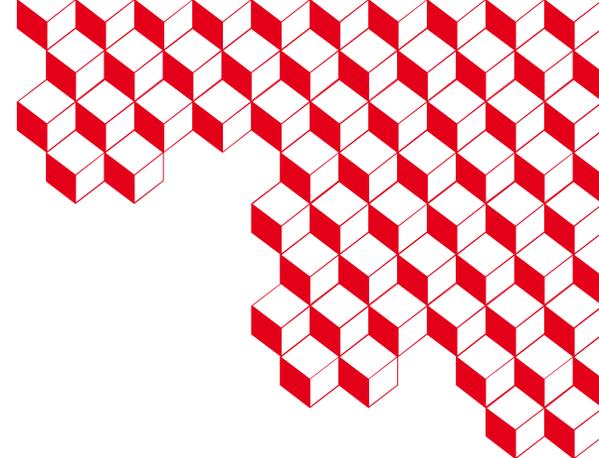
- Un outil souverain pour l'exploitation des machines Exascales
- Compléter la chaîne de valeur du calcul haute performance et pérenniser les développements applicatifs



- Une forte dynamique inter-DO
- Un impact fort sur les programmes du CEA et sur de nombreux défis sociétaux



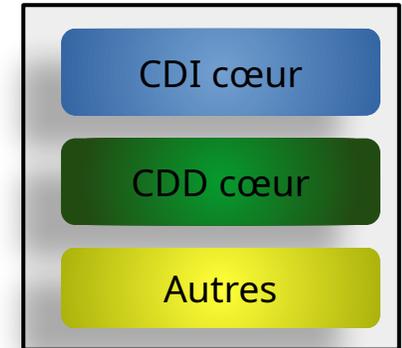
- Une dynamique à rejoindre



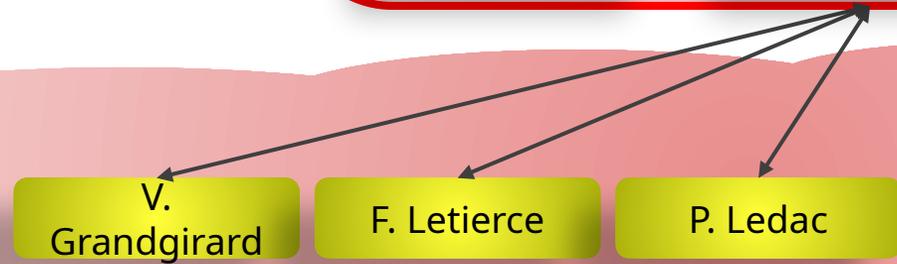
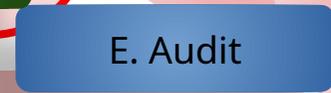
# Merci

CEA SACLAY  
91191 Gif-sur-Yvette Cedex  
France  
[julien.bigot@cea.fr](mailto:julien.bigot@cea.fr)  
Standard. + 33 1 69 08 60 00

# L'équipe



Kokkos  
Comité C++  
Collaborations DoE  
...



Réseau HPC@CEA

# L'équipe cœur



**Fabien BALIGAND**

*SCRUM Master*

DRT



**Ansar Calloo**

*Animateur groupe*

DES



**Mathieu Lobet**

*Animateur groupe*

DRF



**Cedric Chevalier**

*Animateur groupe*

DAM

# L'équipe cœur



**François Letierce**

*développeur / porteur code*

DAM



**Thomas Padioleau**

*développeur / architecte*

DRF



**Rémi Baron**

*développeur*

DES



**Marc Pérache**

*référent technique*

DAM

# L'équipe



**Julien Bigot**

*product owner*

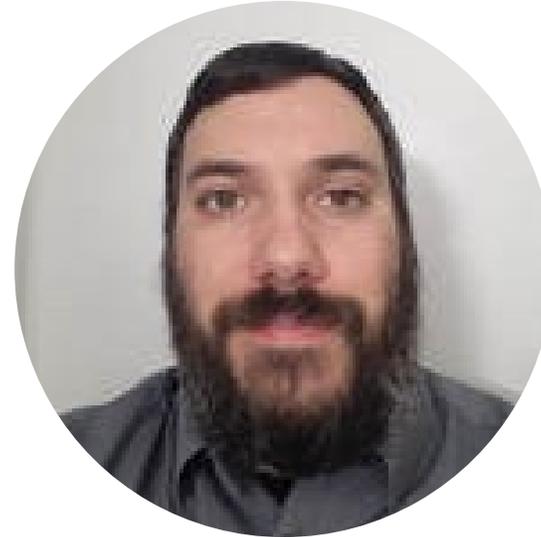
DRF



**Patrick Carribault**

*interface upstream*

DAM



**Julien Jaeger**

*interface upstream*

DAM



**Édouard Audit**

*animateur réseau*

DRF

# L'équipe



**Virgilio Graciano**

*portails de code*

DRS



**Samuel Kokh**

*interface DO*

DES

