

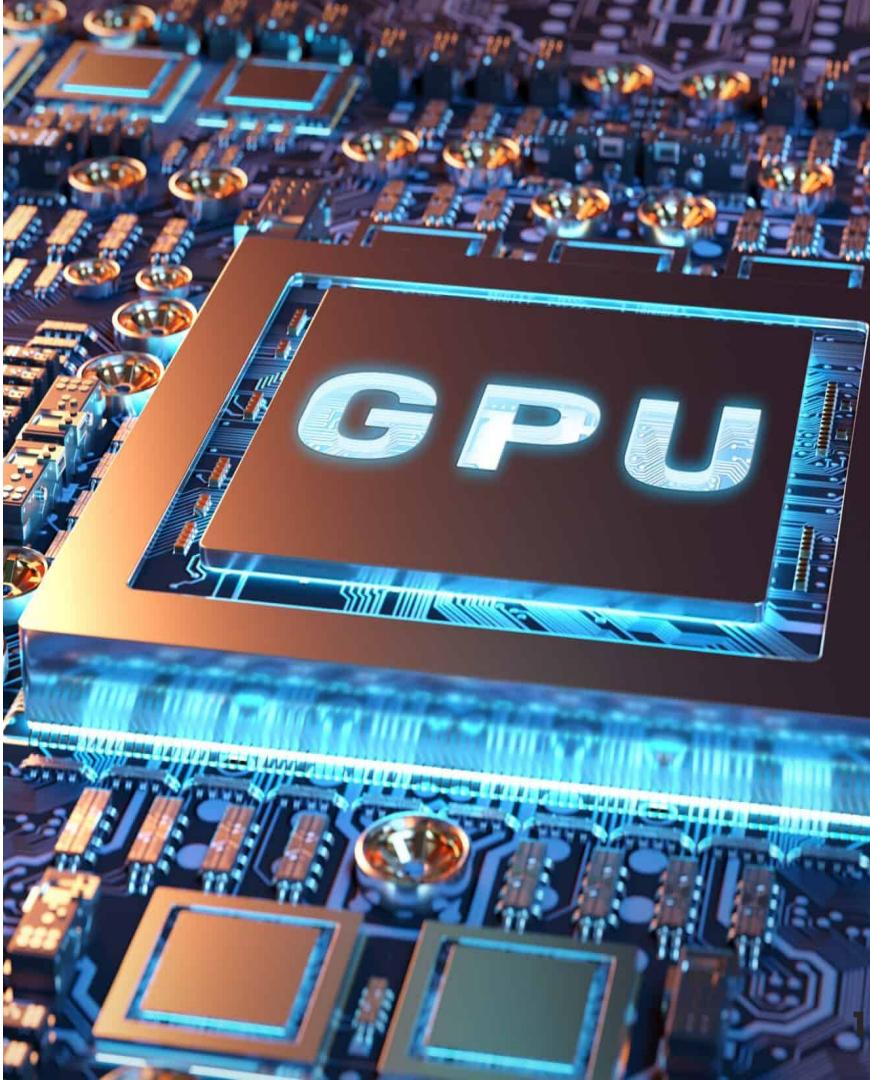


CExA project: Computing at Exascale with Accelerators

Software catalyst for GPU computing



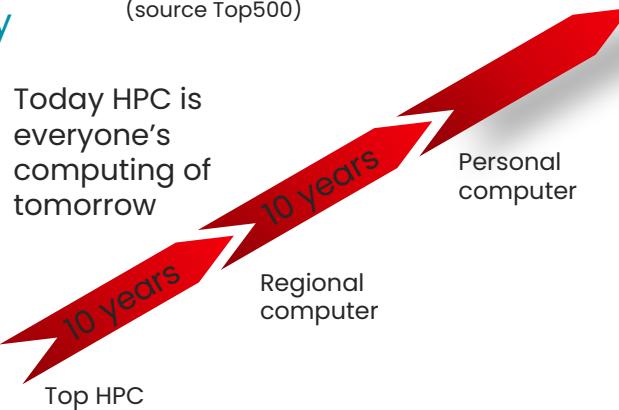
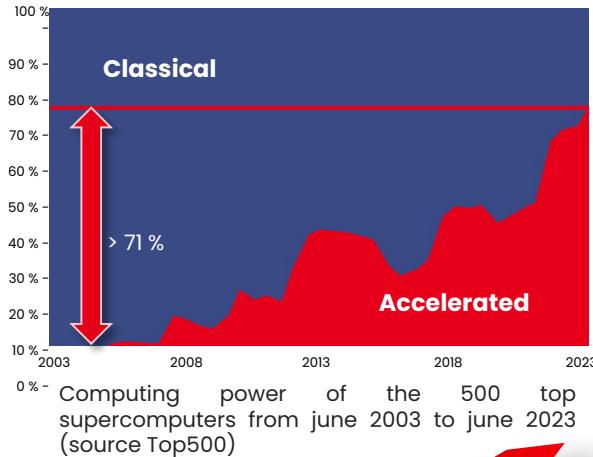
CCRT GPU Workshop – 2 April 2024
Julien Bigot & CExA team





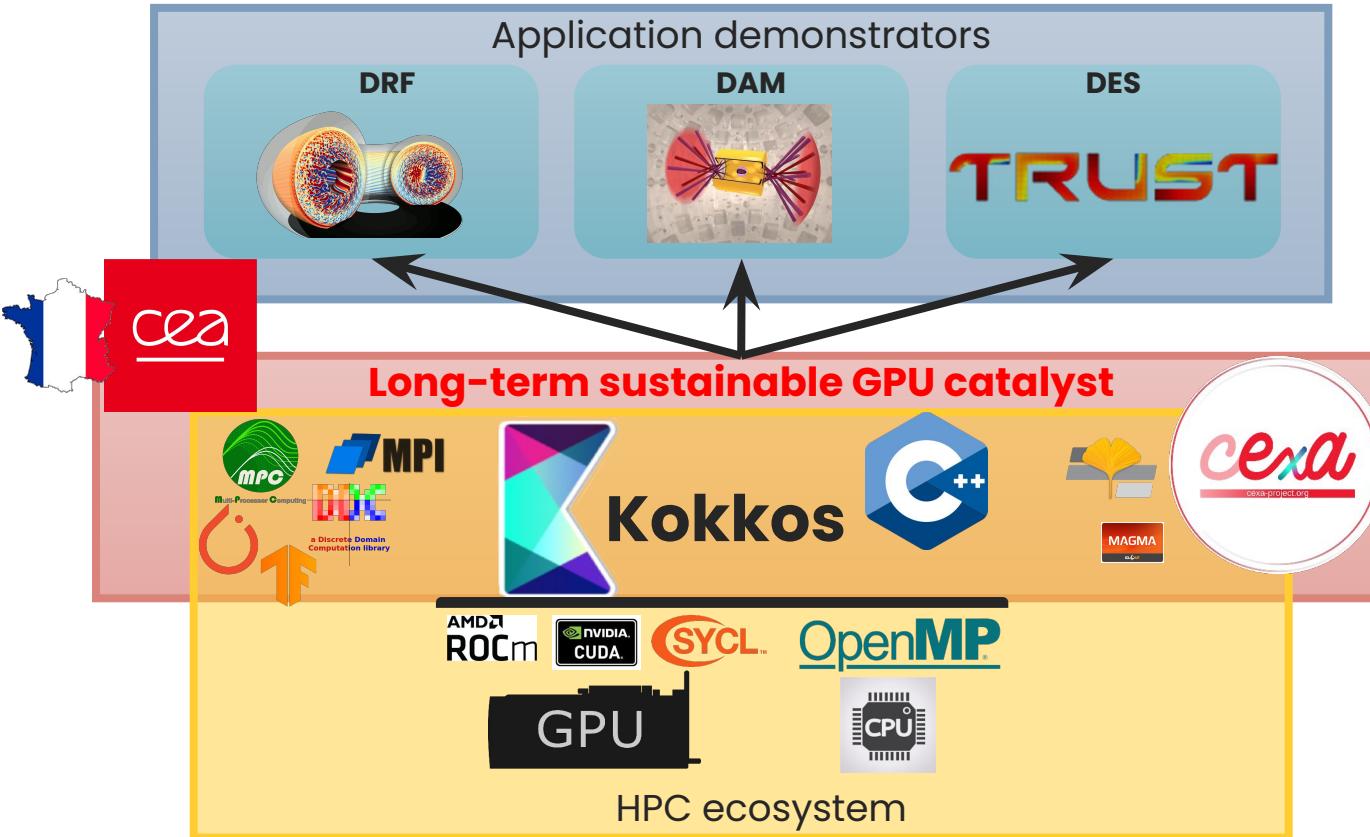
Context

- HPC is a tool largely used **all over CEA**
 - A source of **competitiveness**
- We just entered the **Exascale** era, that means **GPU**
 - European pre-Exascale systems: Mix of AMD & Nvidia
 - First Exascale machines planned in Europe for 2024/2025
 - Jupiter machine at Jülich => Nvidia & Rhea
 - Jules Vernes, French machine at **CEA/TGCC**
 - Need to re-develop applications with **Performance portability**
- GPU middleware: **software catalysts**
 - France and Europe: great research but no production tool
 - App developers are sitting on the back of Buridan's donkey
- A **need** for a long-term sustainable solution
 - **Adapted** to our hardware and software specificities
 - **Trust** in the roadmap





The project



Disseminate
and offer
training in CEA
and at large

Adapt
application
demonstrators

Provide a
long-term
sustainable
software
catalyst for GPU
computing



Available solutions

- Cuda
- HIP
- Kokkos
- OpenACC
- OpenMP (target)
- Raja
- SYCL
 - OneAPI/DPC++
 - AdaptiveC++/OpenSYCL/hipSYCL



Available solutions

- Cuda
- HIP
- Kokkos
- OpenACC
- OpenMP (target)
- **Raja**
- SYCL
 - OneAPI/DPC++
 - **AdaptiveC++/OpenSYCL/hipSYCL**
- **Production grade, with public support**



Available solutions

- Cuda
- HIP
- Kokkos
- OpenACC
- OpenMP (target)
- Raja
- SYCL
 - OneAPI/DPC++
 - AdaptiveC++/OpenSYCL/hipSYCL
- Production grade, with public support
- **Vendor neutral**



Available solutions

- Cuda
- HIP
- Kokkos
- OpenACC
- OpenMP (target)
- Raja
- SYCL
 - OneAPI/DPC++
 - AdaptiveC++/OpenSYCL/hipSYCL
- Production grade, with public support
- Vendor neutral
- Annotations
 - Works best with **imperative languages**: C, Fortran, ...
 - **Compiler integration**: potential for additional optimizations
- Library
 - Suited to language with deep **encapsulation**: C++, ...
 - On top of vendor backends: easier to port to **new hardware**



OpenMP versus Kokkos for a simple GPU loop

```
double* x;
double* y;
double* A;

#pragma omp target data \
    map(to: x[0:Ni]) \
    map(from: y[0:Nj])
{
#pragma omp parallel for
    for (int j = 0 ; j < Nj ; ++j) {
        for (int i = 0 ; i < Ni ; ++i) {
            y[j] += x[i] * A[j*Ni+i];
        }
    }
}
```

```
View<double*, Kokkos::HostSpace> x;
View<double*, Kokkos::HostSpace> y;
View<double*> A;

{
    View<double*> dx;
    deep_copy(dx, x);
    View<double*> dy;
    parallel_for(Nj, KOKKOS_LAMBDA(int j) {
        for (int i = 0 ; i < Ni ; ++i) {
            dy(j) += dx(i) * A(j,i);
        }
    });
    deep_copy(y, dy);
}
```

Ease of use does not offer a clear selection criterion



Available solutions

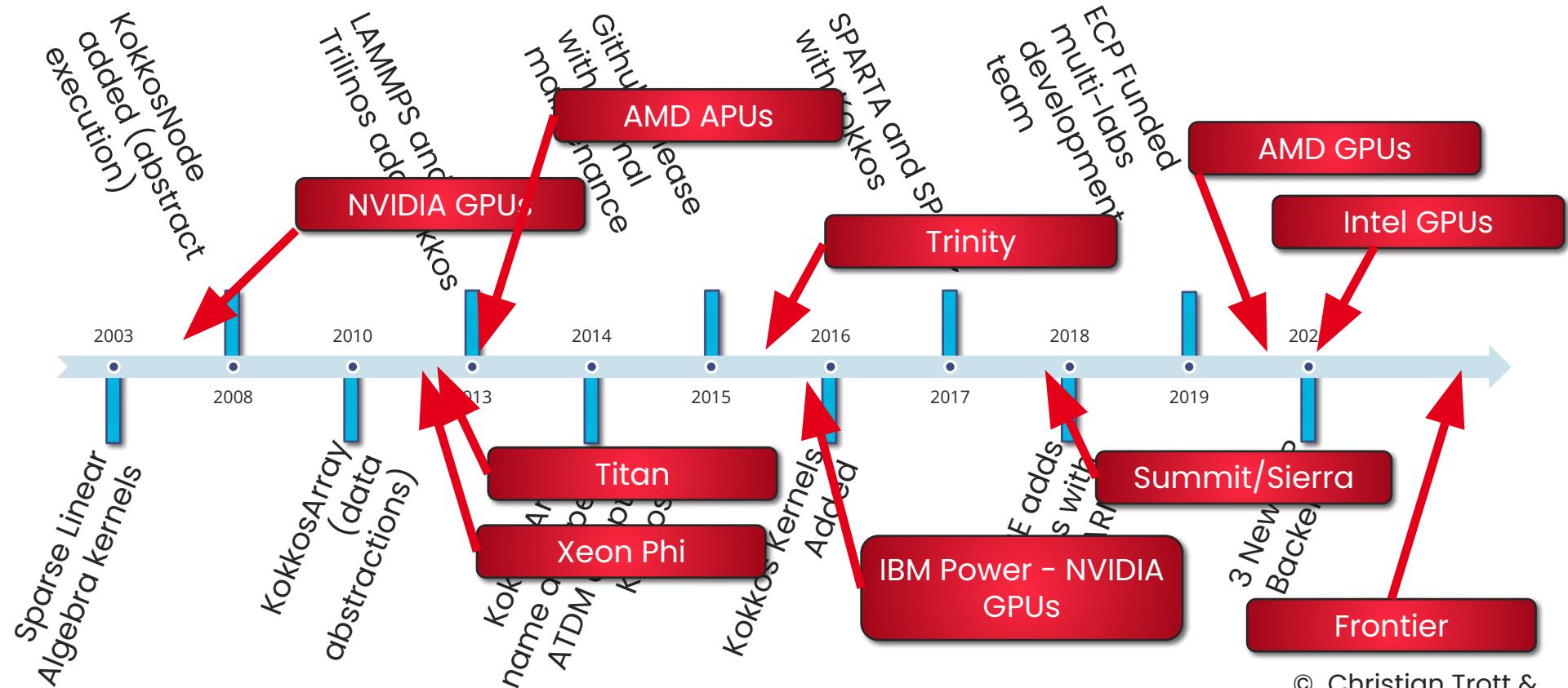
- Cuda
- HIP
- Kokkos
- OpenACC
- **OpenMP (target)**
- Raja
- SYCL
 - OneAPI/DPC++
 - AdaptiveC++/OpenSYCL/hipSYCL

Even with annotations, a code must be redesigned from scratch to go to GPU:
code organization, kernel loop nests,
data structures

- Production grade, with public support
- Vendor neutral
- Annotations
 - Works best with **imperative languages**: C, Fortran, ...
 - Compiler integration: potential for additional optimizations
- **Library**
 - Suited to language with **encapsulation**: C++, ...
 - On top of vendor backends: easier to port to **new hardware**



Kokkos Timeline: not a newcomer

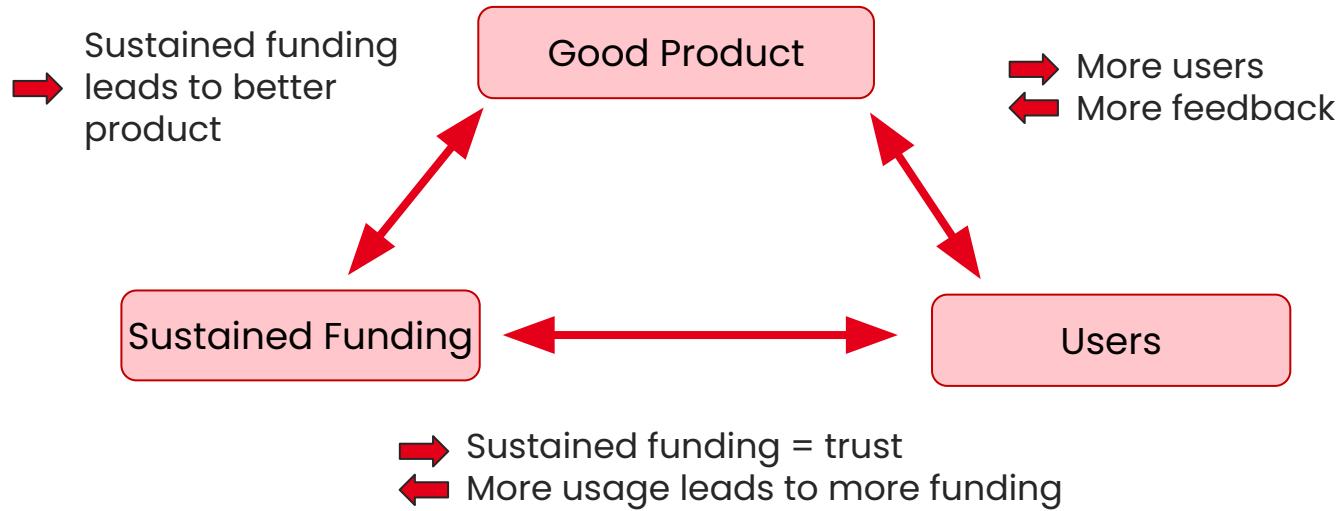


© Christian Trott &
Damien Lebrun Grandjean

2 April 2024



Kokkos Sustainment: a self-reinforcing circle



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

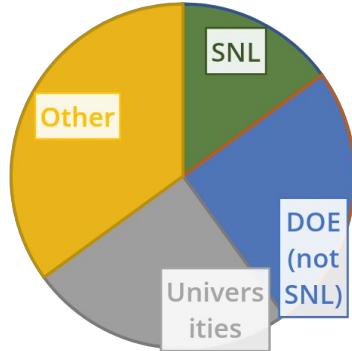


Kokkos Community

Kokkos Slack

<https://kokkosteam.slack.com>

- ~1500 Registered Users
- >150 Institutions
 - Including 34 European



Kokkos Developers



Sandia
National
Laboratories



OAK RIDGE
National Laboratory



Argonne
NATIONAL LABORATORY



Los Alamos
NATIONAL LABORATORY
EST. 1943



BERKELEY LAB



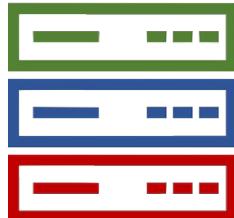
CSCS

Applications and Libraries

- Estimated 150-300 HPC projects using Kokkos
- On the order of three-dozen apps run science and engineering production runs with Kokkos
 - Many apps use multiple Kokkos based libraries
- Similar distribution as the Slack User

© Christian Trott &
Damien Lebrun Grandjean





HIGH PERFORMANCE SOFTWARE FOUNDATION



Hewlett Packard
Enterprise



Lawrence Livermore
National Laboratory



Sandia
National
Laboratories



BERKELEY LAB



Tentative Founding members





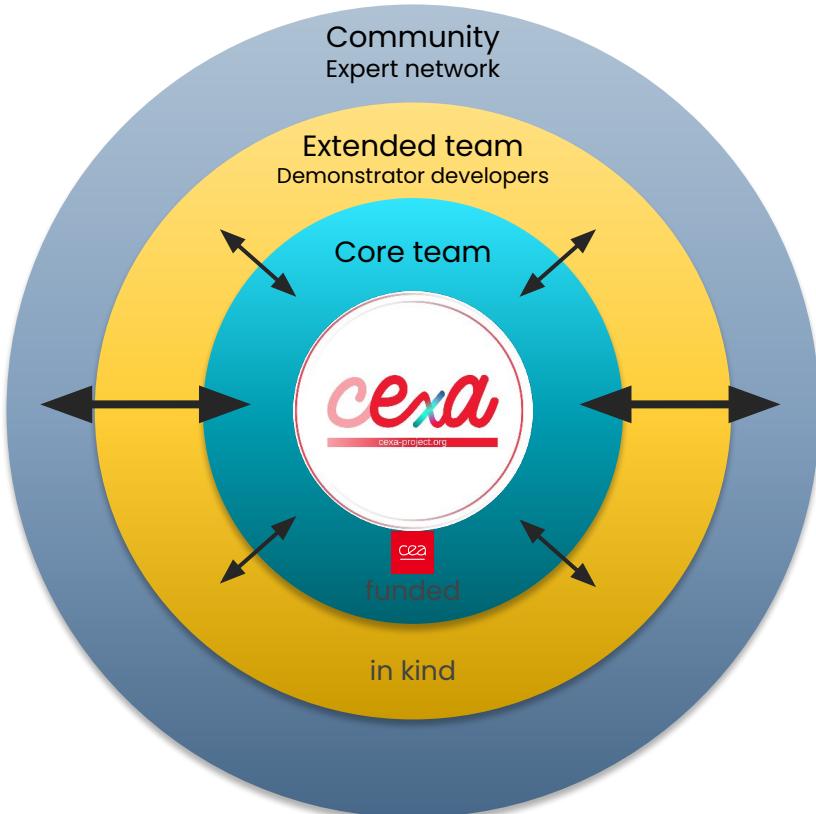
CExA in short

“adopt and adapt” strategy based on  kokkos

- Kokkos : a **strong basis**
 - A software architecture **ready for the future**
 - Performance portability over a wide range of Hardware
 - Mature, free, libre, and open-source
 - An **independent foundation** to own the product
 - Under the Linux Foundation
 - A **standardisation** effort in **ISO C++**
 - A **stepping stone** one step ahead toward **parallel C++**
 - Some **adaptations required**
 - For European **hardware**
 - There is no real hardware sovereignty without software sovereignty
 - For **applications** from CEA, France and Europe
 - Take our specificities into account



Project organization



■ Core team

- Management, implementation and dissemination
- 8 permanent researchers from all over CEA
- + 3 more hires, looking for 3 more
- + Funding for yet 2 more expected this year

■ Extended team

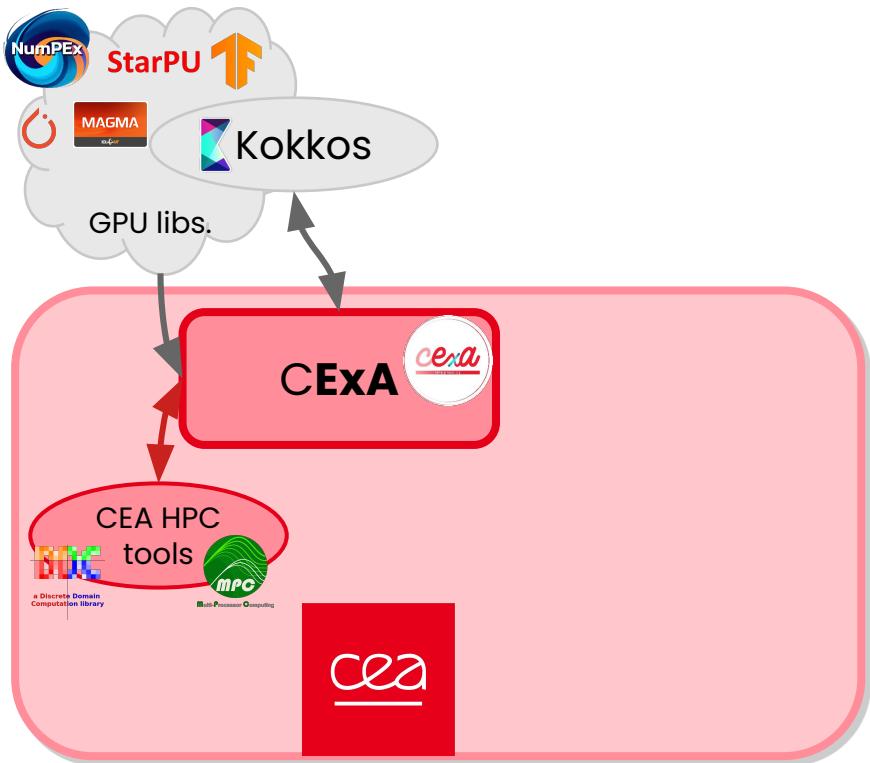
- Demonstrator developers
 - Not funded
 - Find their interest in the participation
- 2 new demonstrators joining this year

■ Community

- Federation of an **expert network**
- Co-design of **CExA**:
 - Identification of needs
 - Usage of **CExA** in applications
- Priority target for **dissemination**
- **Sustainability** of the work

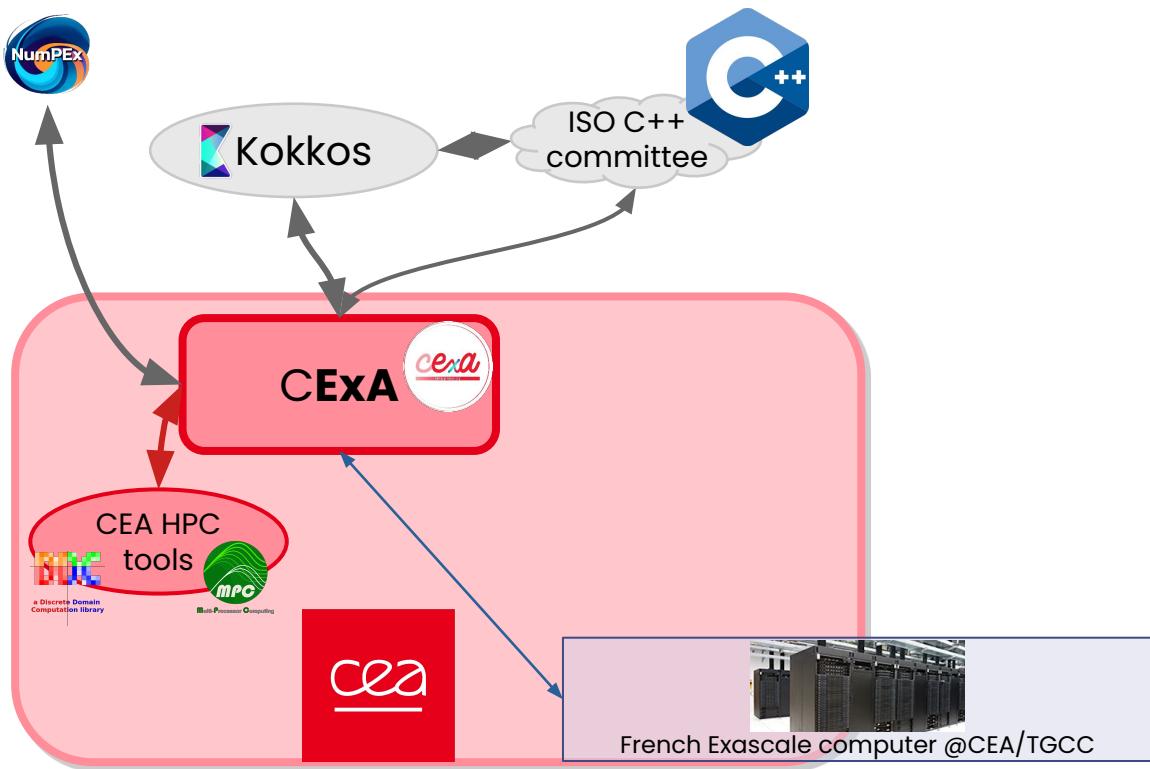


CExA ecosystem: the upstream



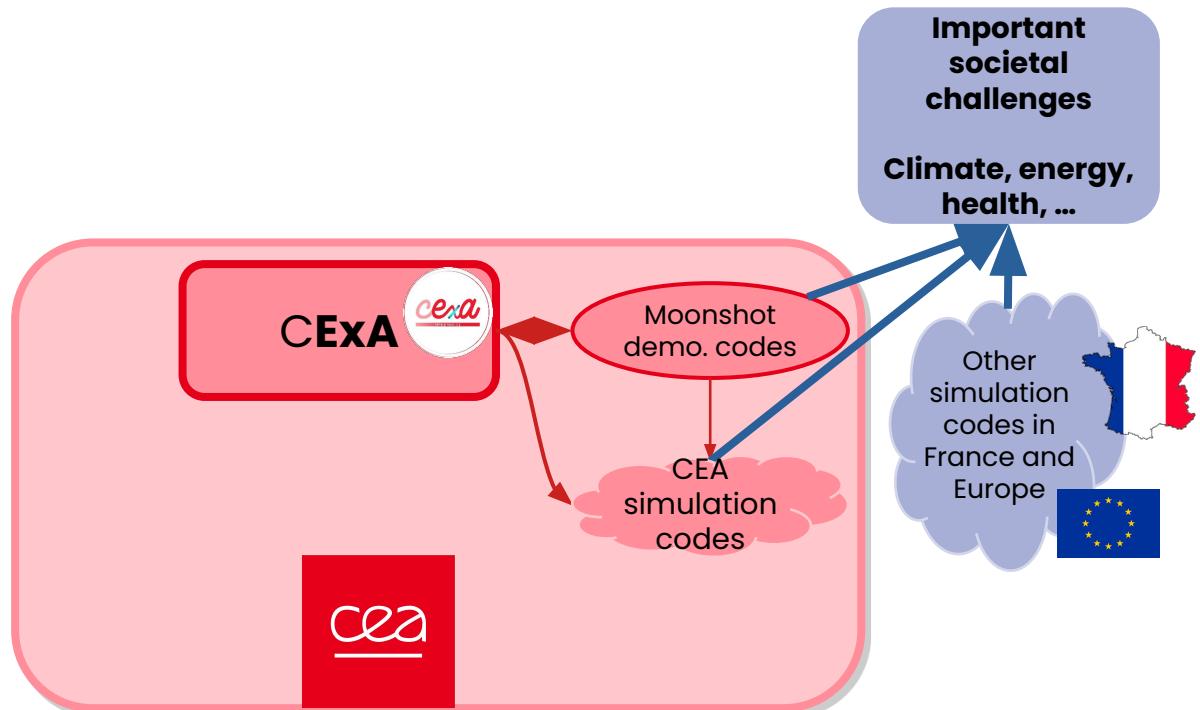
- Kokkos library
 - Our main upstream
 - Kokkos-core but also the whole Ecosystem
- Existing CEA HPC libraries
 - MPC, DDC, Arcane, etc.
 - Integration and exchanges
- Existing GPU libraries
 - Tensorflow, Pytorch, MAGMA, etc.
 - Interfacing thanks to the free, libre, open-source strategy
 - Together with NumPEX PEPR

CExA ecosystem: our partners



- Kokkos & ISO C++ committee
 - A strong relationship
 - Standardisation
 - through Kokkos
 - Normalisation for sustainability of developments
- Jules Vernes project (Exascale France)
 - Strong links established with GENCI, TGCC, and NumPEX
 - Call for proposal soon
 - Including CExA requirements
 - Answer in 2024
 - Selection of the target architecture
 - Delivery in 2025/26
 - CExA ready for production

CExA ecosystem: the downstream



- Two stage downstream
 - The acceleration stage ⇒ applications
 - The final stage ⇒ sociétal challenges
- Integrated demonstrators
 - Co-development
 - Training of the teams
 - Impact in important domains
- CEA applications
 - Training, hackathons, gain of experience
 - A clear choice ⇒ knock-on effect
 - Creation of a community
- FR and EU community
 - Visibility et place of CEA



CExA: what's going on?

- Training
 - First training open to the community last September w. C. Trott & D. Lebrun
 - Next one to come shortly
- Community animation
 - #general-fr slack channel for support, now more than 130 members
 - CExA virtual café twice a month => **Join us** <https://kokkosteam.slack.com>
- Working on our first features
 - Kokkos-FFT (**FFT**), DDC (**Mesh**), Shaman (**Numerical accuracy**)
 - Kokkos-MPI
- Joining the effort of bugfixing, documentation & all the fun!
 - Installation cheatsheet
 - Debugging & optimization for Intel who offered us a platform
 - Improved memory handling on Nvidia Grace Hopper (**Exa1 HE**)
 - Spack recipes maintainers
- Coming next
 - Tools for automating incremental Fortran porting
 - Work on profiling & auto-tuning tools



<https://cexa-project.org/>
<https://github.com/CExA-project>





The Triclade demonstrator (DAM)

Turbulent mixing

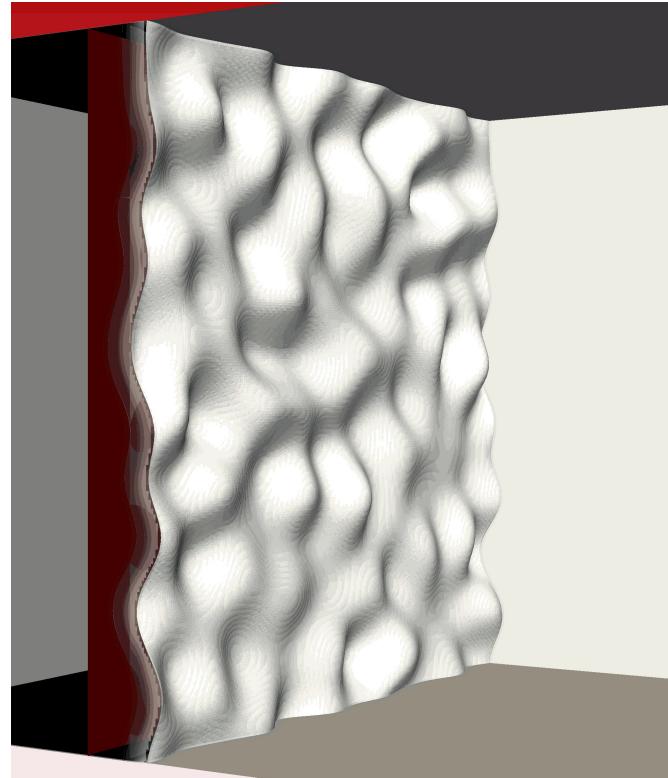
Found in many fields of interest of the CEA:

- Astrophysics ;
- Geophysics ;
- Inertial Confinement Fusion ;
- Etc.

Very complex problem :

- Intrinsically 3D ;
- Multi-scale.

Goal: explore the Kokkos way





The TRUST/TrioCFD demonstrator (DES)

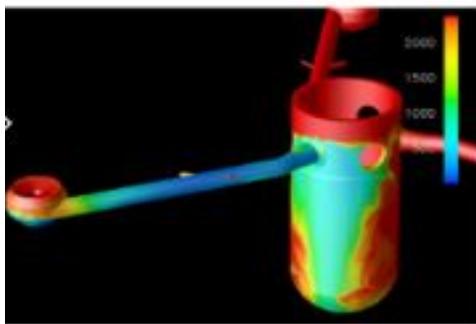


A thermohydraulic platform



A TRUST-based application dedicated to CFD

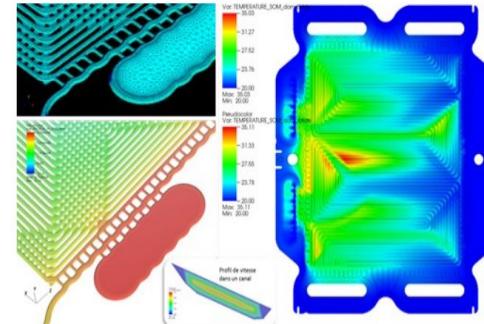
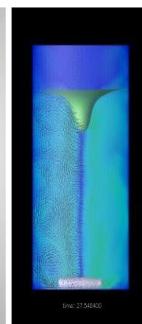
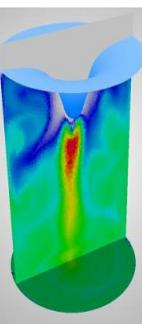
- Fluid mechanics (low/incompressible, mono/multiphase, interface follow)
- For multiple domains



PWR Reactor



Vortex mixer



Fuel cell

- C++, MPI, OpenSource <https://github.com/cea-trust-platform>
- Many other applications based on TRUST: FLICA5, SympyToTRUST, CATHARE3D, Trio-IJK, TrioMC , GENEPI+, PAREX+

Goal: Incremental porting of a huge platform to GPU

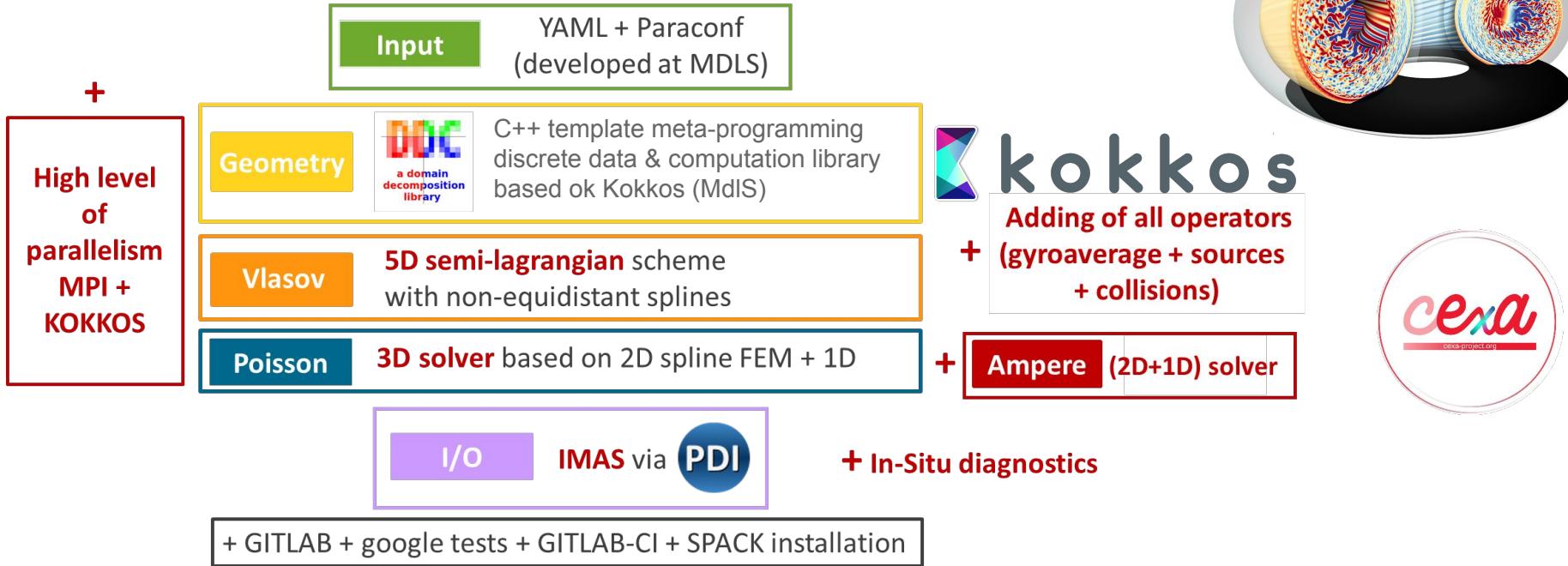




Gysela-X++: towards exascale

- Complete rewriting of the code in modern C++

5D code in modern C++ scalable on exascale architectures



Goal: Rewrite a flagship code for GPU



To conclude



- A **sovereignty** tool to exploit French & EU **Exascale** supercomputers
- **Fill the value chain** of high performance computing and ensure **sustainability** of application developments



- A strong **dynamic** all over the CEA **and beyond**
- A **knock-on effect** with new **synergies** identified every weeks with code developers



- A strong impact on the **programs of CEA** as well as on many **societal challenges**



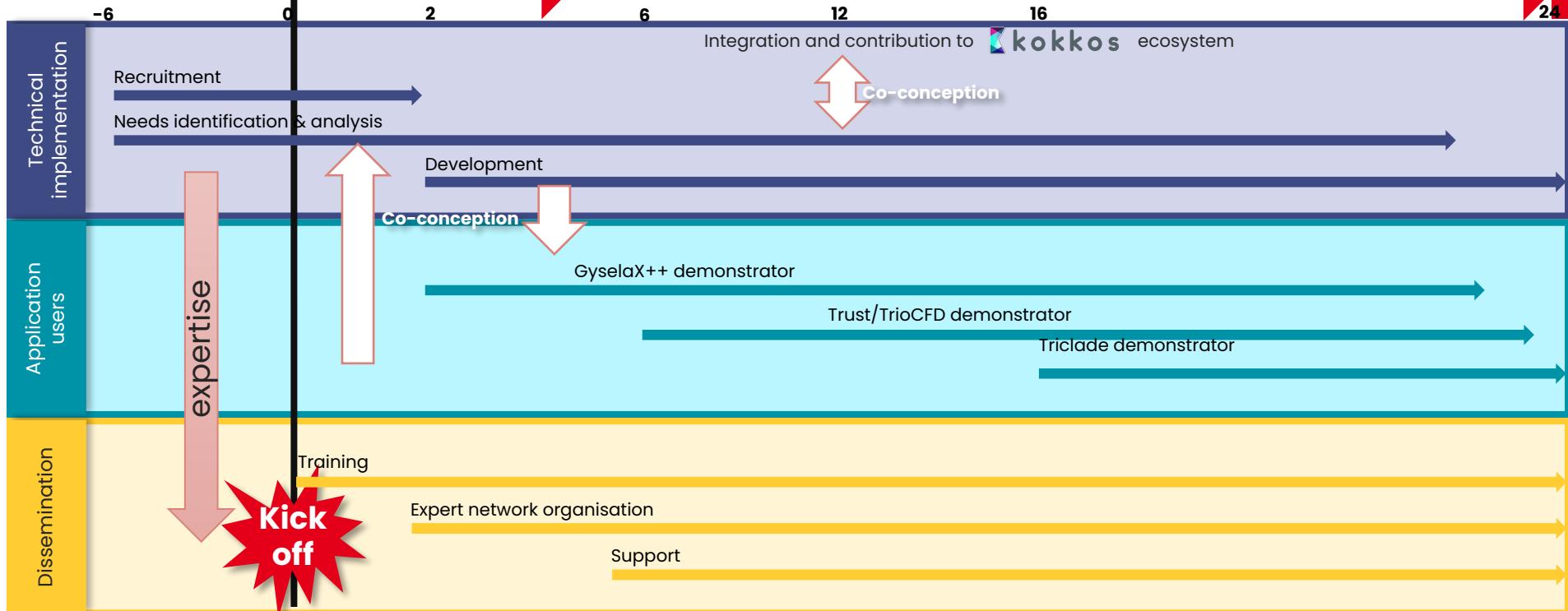
Planning

Preliminary setup

Start: Sept. 2023

Funding: 2 years

End?: Août 2025





The core team

Julien Bigot

product owner



Ansar Calloo

Group leader



Cedric Chevalier

Group leader



Mathieu Lobet

Group leader



Yuuichi Asahi

Senior developer



Thierry Antoun

Developer



Rémi Baron

Senior developer



Thomas Padoleau

Senior developer



Thomas Padoleau

Developer



The extended team

Pierre Ledac

Trust/TrioCFD lead



Virginie Grandgirard

GyselaX++ lead



François Letierce

Triclade lead



Julien Jaeger

DAM link



Édouard Audit

Network animator



Samuel Kokh

DES link



Patrick Carribault

DAM link