ExaFLOW – Enabling Exascale Fluid Dynamics Simulations

Erwin Laure, KTH, Stockholm, SWEDEN

ExaFLOW

→ Address current algorithmic bottlenecks to enable the use of accurate CFD codes for problems of practical engineering interest

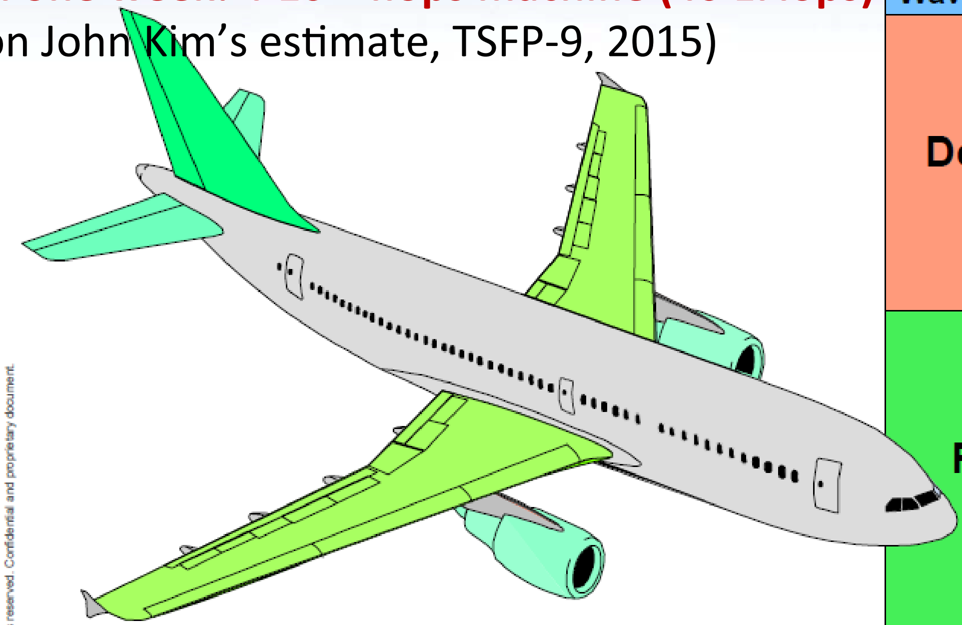
The Airbus Challenge, David Hills, 2008

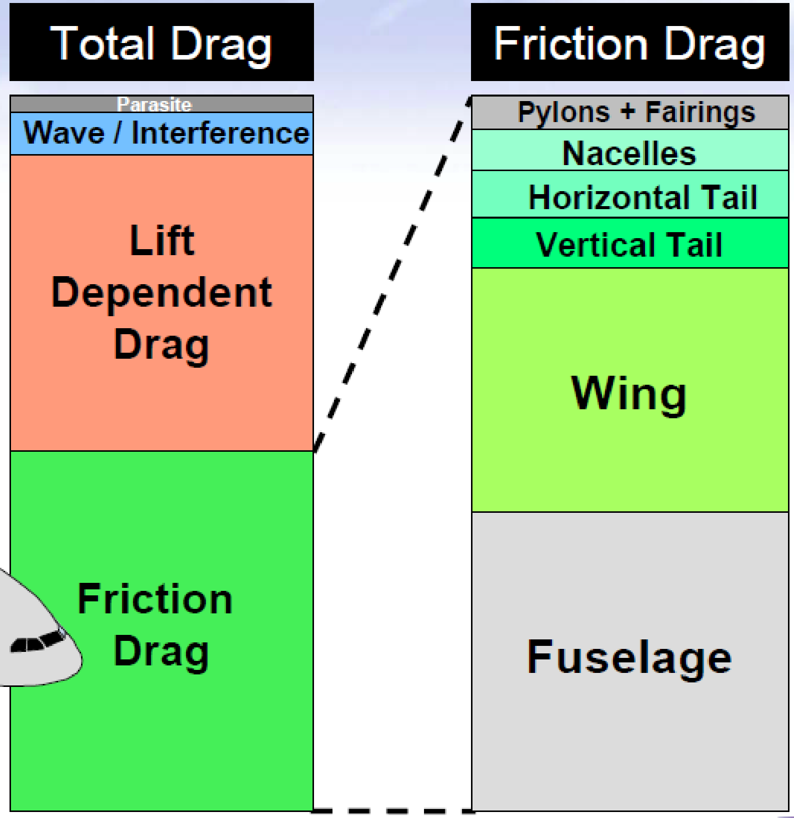
A Brief Diversion Into Aircraft Drag

A world of challenge & opportunity

- Typical break down of overall aircraft† drag by form & component

An Airbus 310 cruising at 250 m/s at 10000m
Teraflops machine (10^{12} Flops): $8 \cdot 10^5$ years
Result in one week: $4 \cdot 10^{19}$ flops machine (40 EFlops)
(based on John Kim's estimate, TSFP-9, 2015)





† = Based on a typical A320

Navier – Stokes equations

Data from Mira (2013), million core hours

• Engineering/CFD	525	19%
• Subsurface flow & reactive transport	80	3%
• Combustion	100	4%
• Climate	280	10%
• Astrophysics	28	1%
• Supernovae	105	4%
	1118	**40%**

(fraction of Navier-Stokes based simulation on current supercomputers)

ExaFLOW Overview

The main goal of the project is to address current algorithmic bottlenecks to enable the use of accurate CFD codes for problems of practical engineering interest. The focus will be on different simulation aspects including:

- accurate **error control** and **adaptive mesh refinement** in complex computational domains,
- **solver efficiency** via mixed discontinuous and continuous Galerkin methods and appropriate optimised preconditioners,
- strategies to ensure **fault tolerance** and resilience,
- **heterogeneous modelling** to allow for different solution algorithms in different domain zones,
- parallel **input/output** for extreme data, employing novel data reduction algorithms (feature-based in-situ analysis),
- **energy awareness** of high-order methods,
- 4 different codes.

ExaFLOW Partners

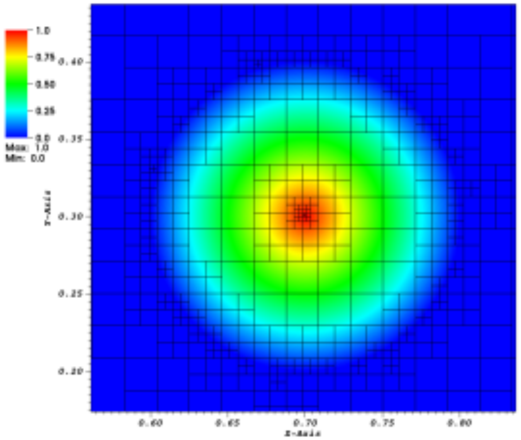
- KTH Stockholm, PDC and Mechanics (Coordinator)
- Imperial College, London, CFD
- University of Southampton, Aerodynamics
- University of Edinburgh, EPCC
- University of Stuttgart, HLRS and Aerodynamics
- EPF Lausanne, Mathematics

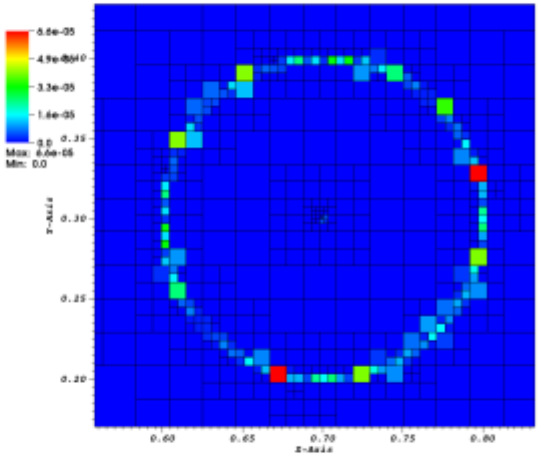
- McLaren Racing, UK
- Automotive Simulation Center Stuttgart

Algorithmic Development

- 3 main objectives to develop the mathematical and algorithmic knowledge needed to tackle key objectives and enable exascale-level CFD software:
  - **Objective 1:** error control through adaption, heterogeneous modelling and resilience
  - **Objective 2:** strong scaling at exascale
  - **Objective 3:** techniques for I/O at exascale

WP1: 6 month progress highlights

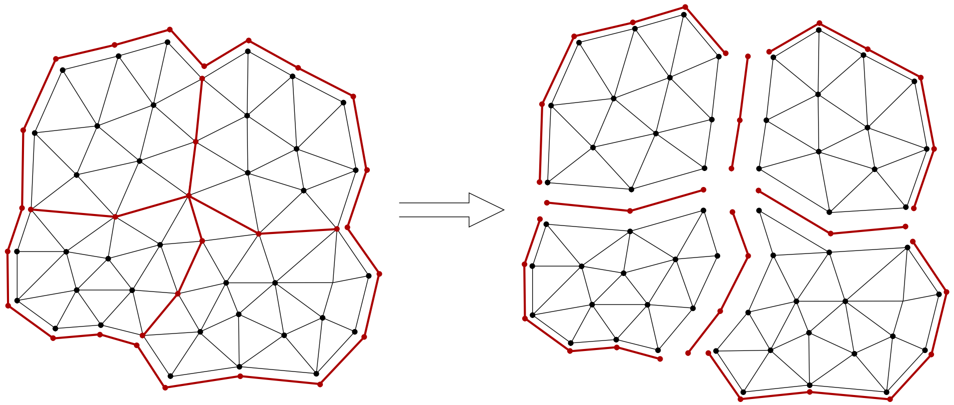




Working implementation of a spectral error estimator to drive a *h*-adaptive process for Objective 1 **(KTH)**

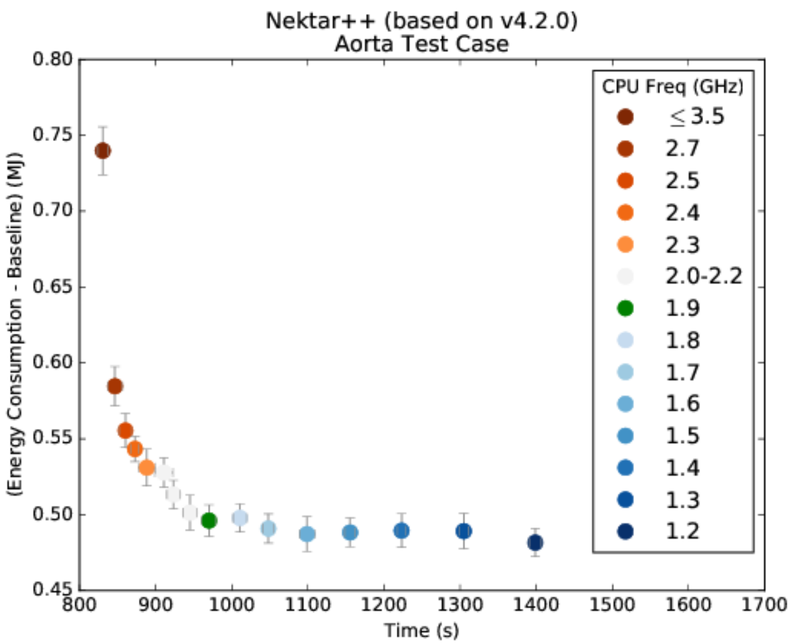
Initial formulation of hybrid CG-HDG on one node undergoing initial testing for Objective 2, aim to reduce communication bottleneck **(ICL)**

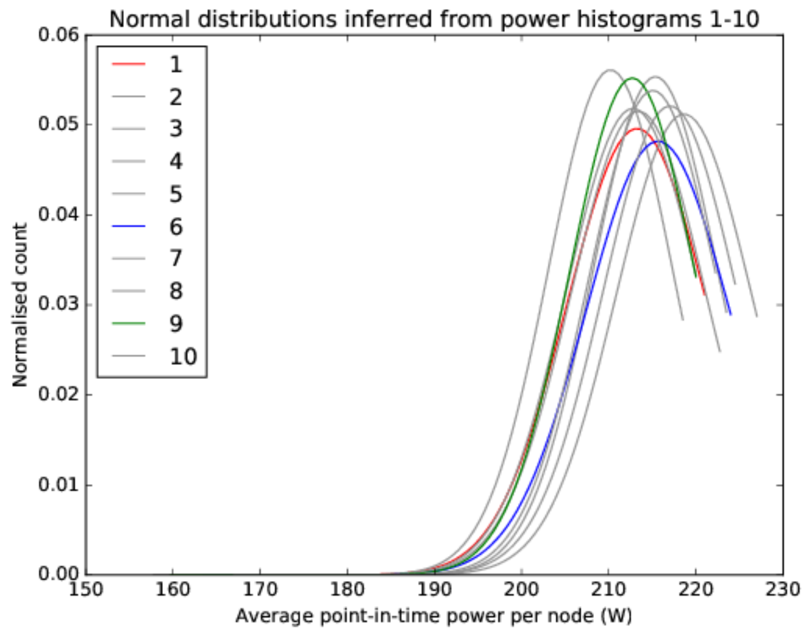
Preprint of investigation into resilience measures for hard/soft errors during runtime for Objective 1 **(EPFL)**



Power and energy profiling to build baselines

- Building power and energy profiles of Nektar++ and Nek5000 to establish baseline upon which to measure improvements.
- Using full test cases to exercise complete code.
- Energy-to-solution and time-to-solution as a function of processor frequency (left); looking for optimal trade-offs.
- Variation seen between otherwise identical runs of the same code and test-case (right); looking to quantify uncertainty in measurements.

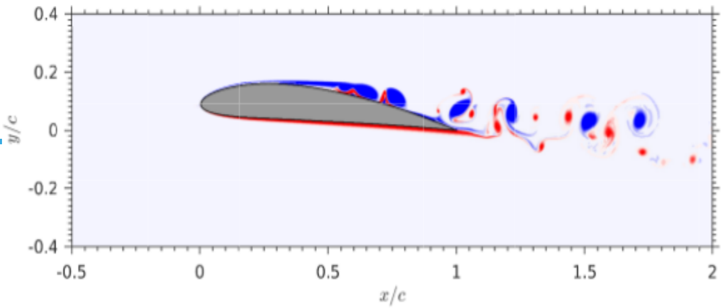




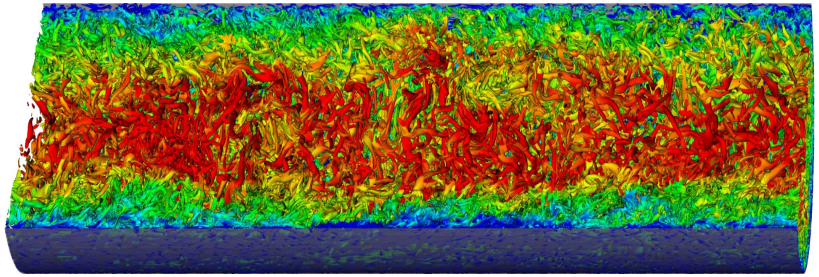
Adding modern parallel I/O to Nektar++

- Code bottleneck found in I/O.
  - Slows checkpoint / restart and final results writes.
- Previously multiple files, per process writing.
- Now using HDF5 atop MPI-IO for parallel I/O to single file.
  - Non-trivial mapping of elements on each process to file.
  - Allows checkpoint on X nodes, restart on Z nodes; X!=Z.
  - Fit problem to available resources, even when solution requires >1 jobs of different lengths/widths.
- Builds on top of work done in prior project, now ready to merge into trunk.
- Gives implementation to bring to other project codes doing similar I/O with non-trivial mappings.

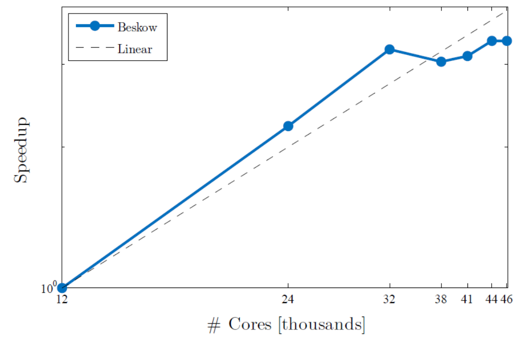
Pilot Cases



Five computationally-demanding use cases suitable for demonstrating the need for exascale capabilities have been created:

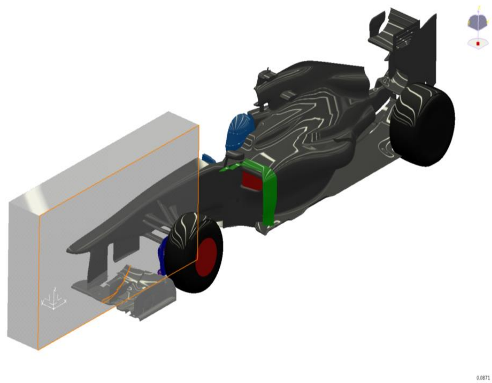


- **NACA4412 (compressible)** - Soton
- **NACA4412 (incompressible)** - KTH
- **Jet in crossflow** - KTH/Stuttgart
- **Automotive/flow past a car** - ASCS
- **Imperial Front wing** – Imperial/McLaren



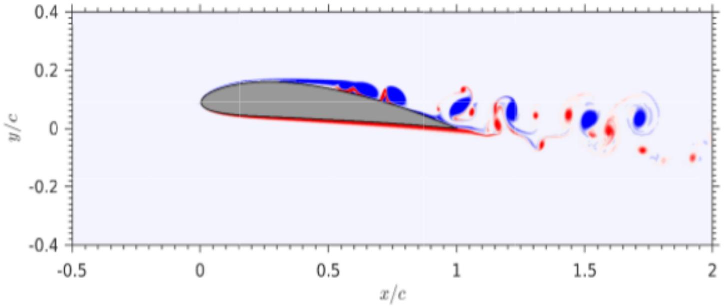
Some of these have already been published online on the ExaFLOW project website. The others will soon be made available via Deliverable 3.1.

Quantitative measures to ensure correct flow physics is reproduced after code optimisations have been defined.

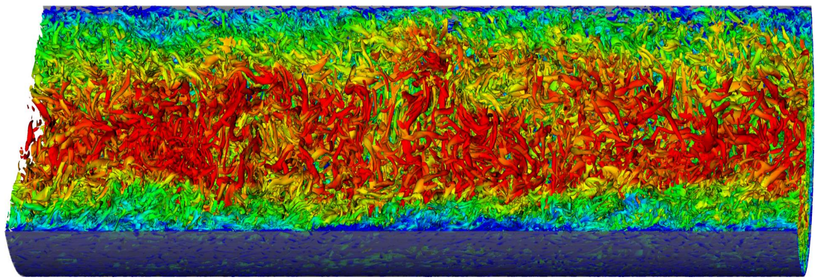


Computational requirements have been investigated and internal evaluation of use cases is under way.

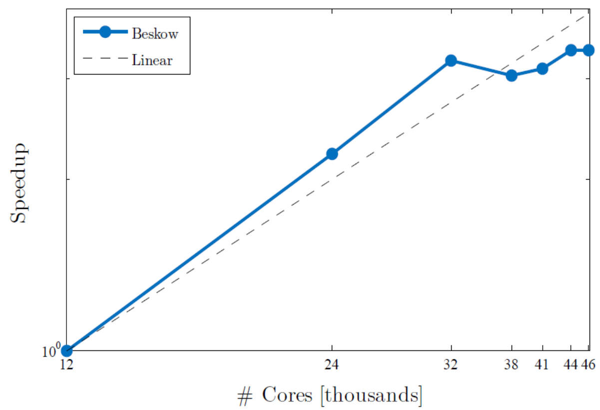
Vortices being generated from a NACA 4412 airfoil:



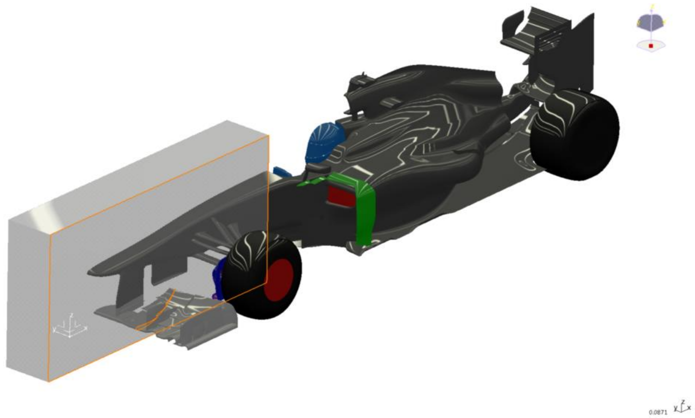
Turbulent straight pipe simulation (part of the jet in cross flow case):



Preliminary results of strong scaling from the incompressible NACA-4412 case (3.2 billion solution points):

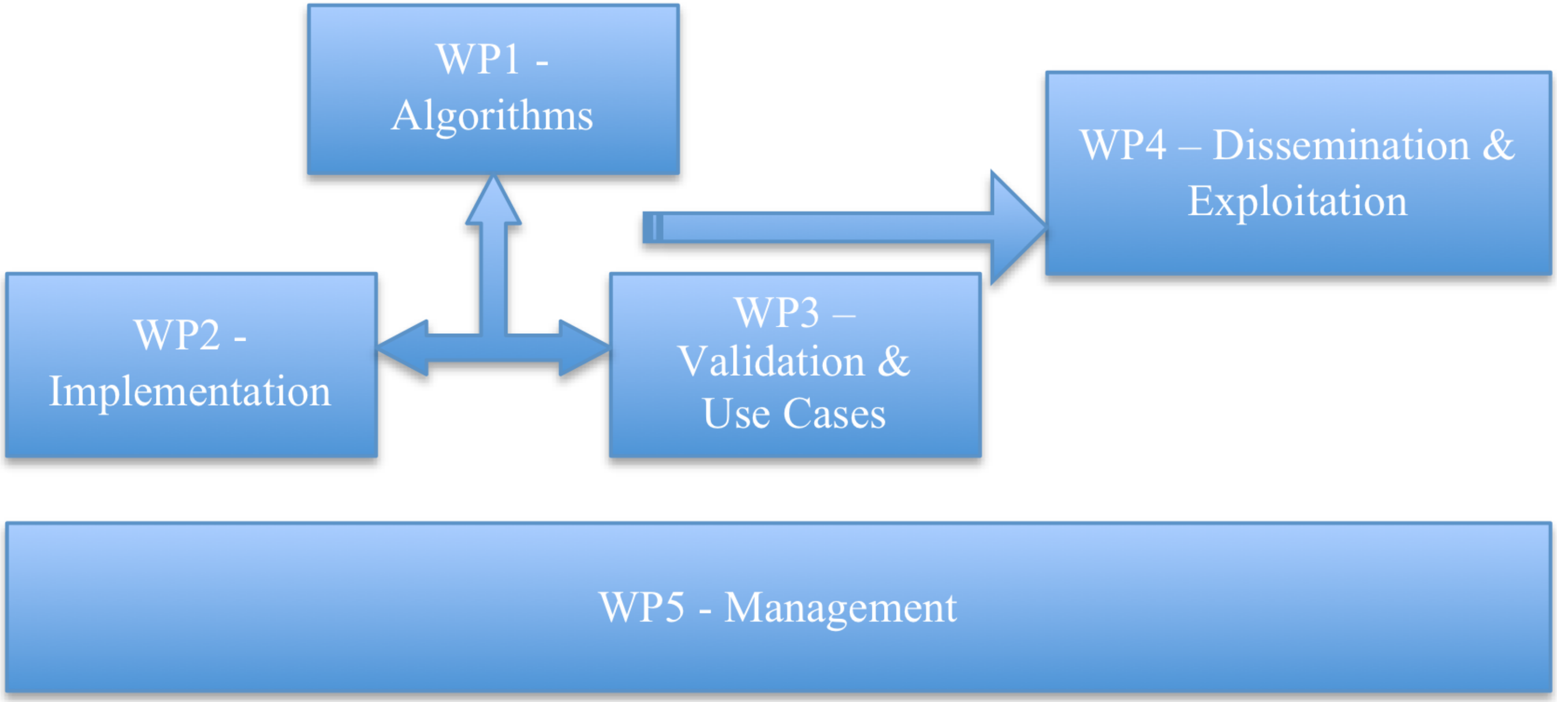


Geometry for Imperial front wing use case:



Work Packages

1. Algorithmic improvements towards exascale (IC)
2. Efficiency improvements towards exascale (UEDIN)
3. Validation & use cases (SOTON)
4. Dissemination and exploitation (USTUTT)
5. Management (KTH)



Conclusions and Outlook

- Fluid mechanics is a **prime example** for exascale
- ExaFLOW will address some of the issues when it comes to **practial applications**
  - error control and adaptive meshing for larger and more complex simulation domains; capable of dynamic remeshing if necessary.
  - Heterogeneous modelling
  - Resilience & fault tolerance
  - data handling, complex feature extraction (in-situ) and sharing of simulation data.
- Enhancing **community codes** (Nek5000, Nektar++, SBLI); Open-source development of all components