# Benefits and Challenges of Software Development Kits with Focus on xSDK

**Ulrike Meier Yang**

**June 8, 2023**

**PESO Workshop**

**Lawrence Livermore National Laboratory**

# What is a software development kit?

- Better definition for ECP:

  **A group of independent but interrelated software products comprising a unified whole**
  **or**
  **an ecosystem of related software products**

- operating system. (techterms.com)

# ECP SDKs (PI)

- Programming Models and Runtimes (Sameer Shende)

- Development Tools (Bart Miller)

- Math Libraries (xSDK) (Ulrike Yang)

- Data and Visualization Software (Patrick O'Leary)

- Workflows (Dan Laney)

- All SDK members contained in E4S (Sameer Shende)

# History of xSDK

**xSDK history:** Work began in ASCR/BER partnership, Sept 2014, IDEAS project (Heroux, McInnes, Moulton)

Needed for BER multiscale, multiphysics integrated surface-subsurface hydrology models
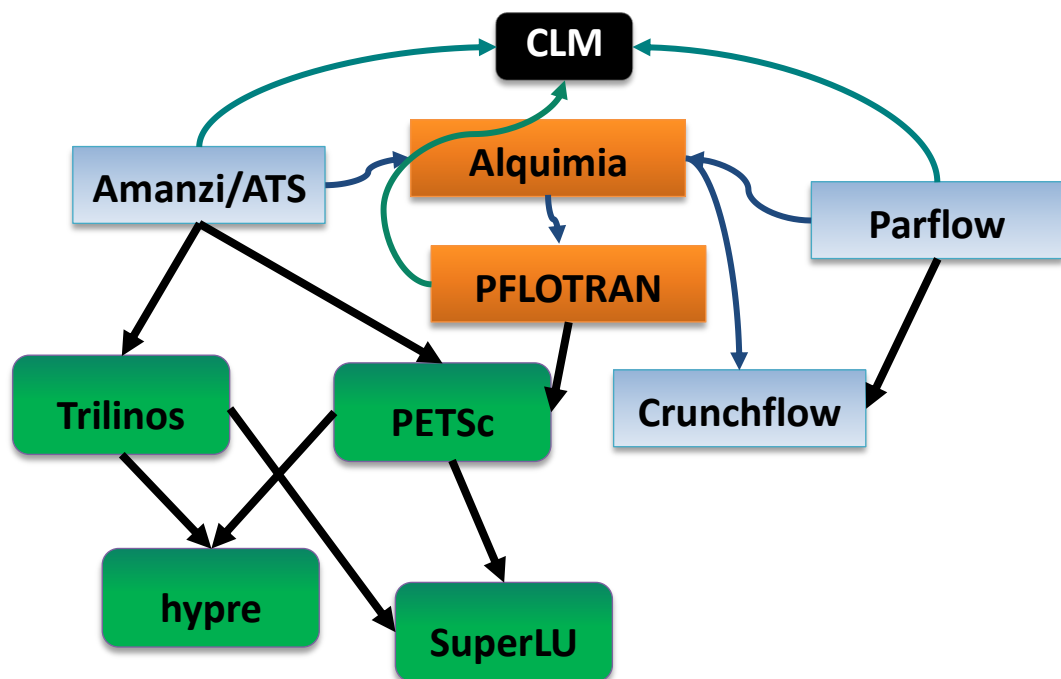


## Next-generation scientific simulations require combined use of independent packages

- **Prior to xSDK effort, many difficulties to build required libraries into a single executable due to many incompatibilities**

- Installing multiple independent software packages is tedious and error prone
  - Need consistency of compiler (+version, options), 3rd-party packages, etc.
  - Namespace and version conflicts make simultaneous build/link of packages difficult

- Multilayer interoperability among packages requires careful design and sustainable coordination
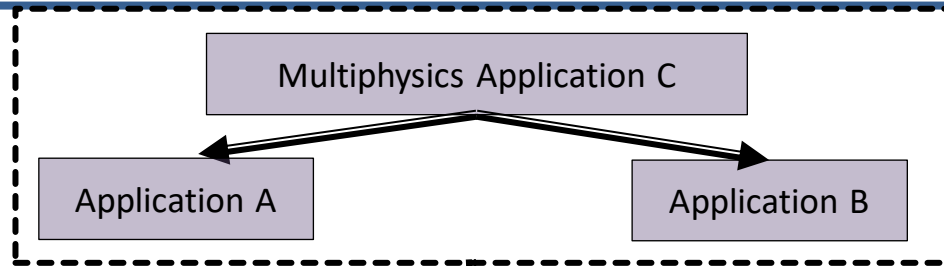
Lawrence Livermore National Laboratory
LLNL-PRES-849915

CASC

xSDK

ECP EXASCALE COMPUTING PROJECT

NNSA National Nuclear Security Administration

4

# xSDK History: Version 0.1.0: April 2016



https://xsdk.info

Notation: **A** ➡ **B**:
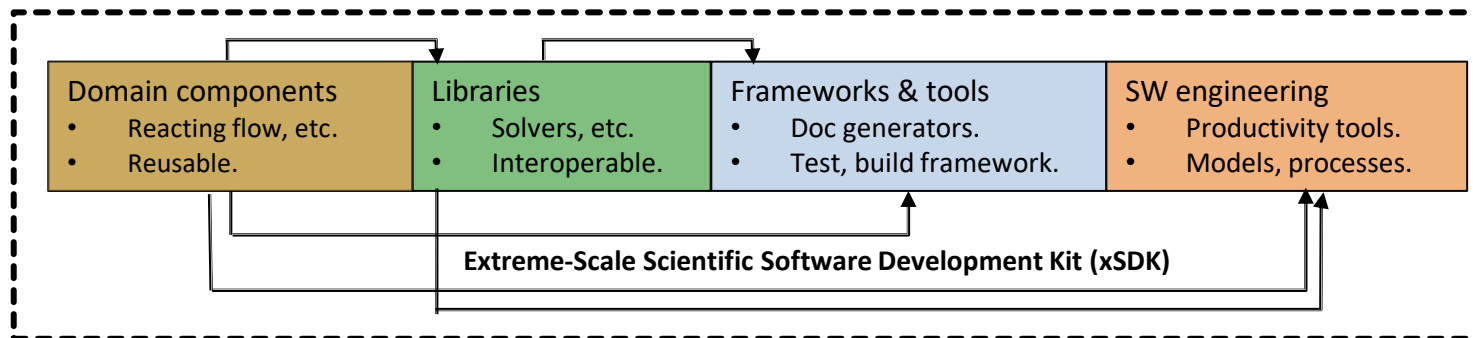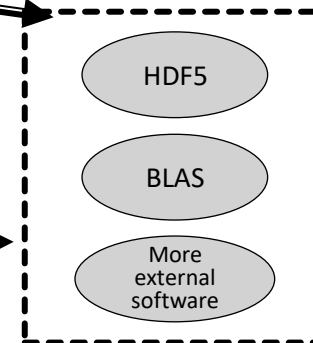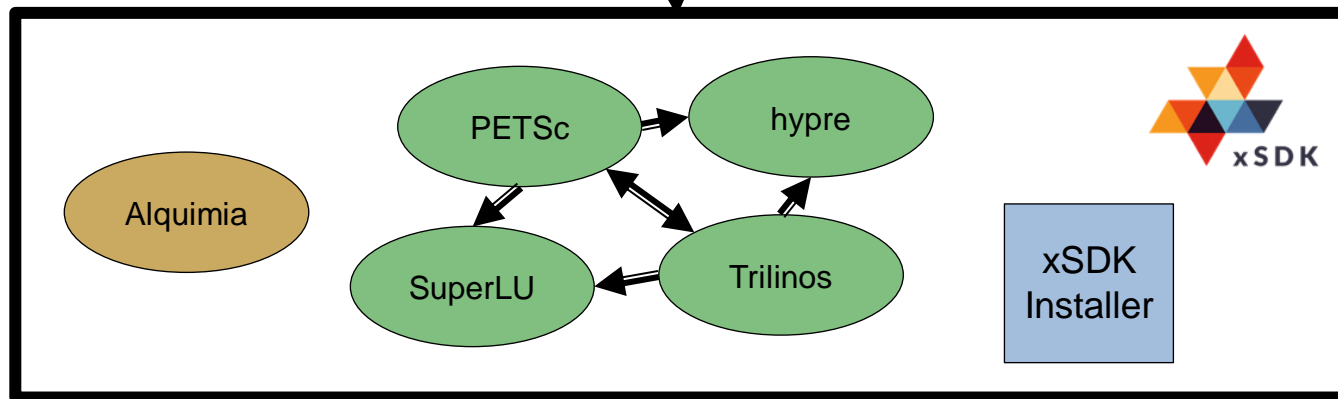**A** can use **B** to provide functionality on behalf of **A**

**April 2016**
- 4 math libraries
- 1 domain component
- PETSc-based xSDK installer
- 14 mandatory xSDK community policies

xSDK functionality, April 2016
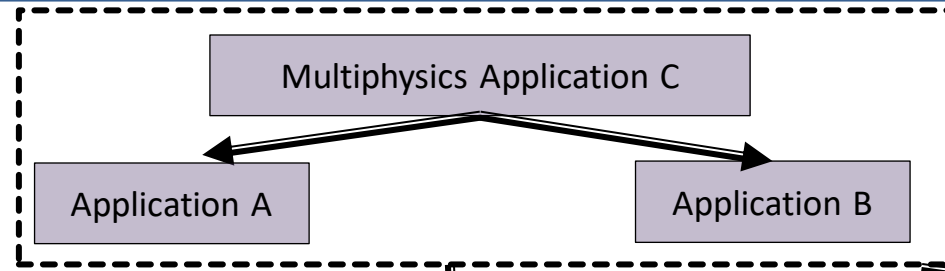
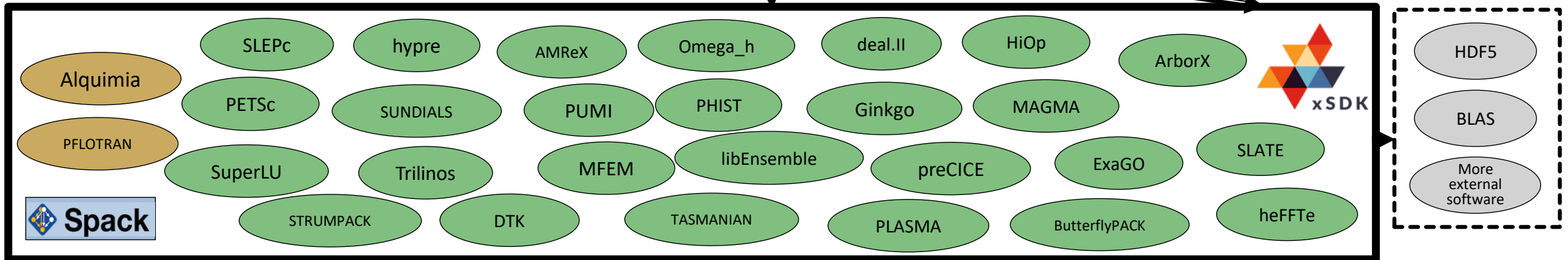Tested on key machines at ALCF, NERSC, OLCF, also Linux, Mac OS X

Multiphysics Application C

Application A

Application B

PETSc

hypre

Alquimia

SuperLU

Trilinos

xSDK Installer

HDF5

BLAS

More external software

Domain components
- Reacting flow, etc.
- Reusable.

Libraries
- Solvers, etc.
- Interoperable.

Frameworks & tools
- Doc generators.
- Test, build framework.

SW engineering
- Productivity tools.
- Models, processes.

Extreme-Scale Scientific Software Development Kit (xSDK)

Lawrence Livermore National Laboratory
LLNL-PRES-849915

CASC

xSDK

ECP EXASCALE COMPUTING PROJECT

NNSA National Nuclear Security Administration

5

# xSDK Version 0.8.0: November 2022

https://xsdk.info

# xSDK Libraries



- **AMReX**: Ann Almgren, Michele Rosso (LBNL)
- **DTK**: Stuart Slattery, Bruno Turcksin (ORNL)
- **deal.II**: Wolfgang Bangerth (Colorado State University)
- **ExaGO:** S. Abhyankar (PNNL)
- **Ginkgo**: Hartwig Anzt (Karlsruhe Institute of Technology)
- **heFFTe:** Stan Tomov (UTK)
- **HiOp:** C. Petra (LLNL)
- **hypre**: Rob Falgout, Ulrike Yang (LLNL)
- **libEnsemble**: Stefan Wild, Steve Hudson (ANL)
- **MAGMA** and **PLASMA**: Piotr Luszczek (UTK)
- **MFEM**: Aaron Fischer, Tzanio Kolev (LLNL)
- **Omega_h**: Dan Ibanez (SNL)
- **PETSc/TAO**: Satish Balay, Alp Denner, Barry Smith (ANL)
- **preCICE:** Frederic Simonis (Technical University Munich)
- **PUMI**: Cameron Smith (RPI)
- **SUNDIALS**: Cody Balos, David Gardner, Carol Woodward (LLNL)
- **SuperLU, STRUMPACK, ButterflyPACK**: Sherry Li, Pieter Ghysels, Yang Liu (LBNL)
- **TASMANIAN**: Miroslav Stoyanov, Damien Lebrun Grandie (ORNL)
- **Trilinos**: Keita Teranishi, Jim Willenbring, Sam Knight (SNL)
- **PHIST**: Jonas Thies (DLR, German Aerospace Center)
- **SLEPc**: José Roman (Universitat Politècnica de València)

# Technical Challenges

- Dealing with Different Strategies of Individual Components
  - Solution: **xSDK community policies,** which help address challenges in interoperability and sustainability of software developed by diverse groups at different institutions
    **https://github.com/xsdk-project/xsdk-community-policies**

---

**xSDK compatible package: must satisfy the mandatory xSDK policies (M1, ..., M17)**
Topics include configuring, installing, testing, MPI usage, portability, contact and version information, open-source licensing, namespacing, and repository access

Also specify **recommended policies**, which are encouraged but not required (R1, ..., R8)

Topics include public repository access, error handling, freeing system resources, and library dependencies, documentation quality

---

**xSDK member package**:
(1) Must be an xSDK-compatible package, *and*
(2) it uses or can be used by another package in the xSDK, and the connecting interface is regularly tested for regressions.

# xSDK community policies

**https://github.com/xsdk-project/xsdk-community-policies**

**https://xsdk.info/policies**

## Mandatory xSDK policies: must be satisfied

**M1.** Support portable installation through Spack
(includes xSDK Spack variant guildelines)

**M2.** Provide a comprehensive test suite.

**M3.** Employ user-provided MPI communicator.

**M4**. Give best effort at portability to key architectures.

**M5.** Provide a documented, reliable way to contact the development team.

**M6.** Respect system resources and settings made by other previously called packages.

**M7.** Come with an open-source license.

**M8.** Provide a runtime API to return the current version number of the software.

**M9.** Use a limited and well-defined symbol, macro, library, and include file name space.

**M10.** Provide publicly available repository.

**M11.** Have no hardwired print or IO statements.

**M12.** Allow installing, building, and linking against an outside copy of external software.

**M13.** Install headers and libraries under <prefix>/include/ and <prefix>/lib/.

**M14.** Be buildable using 64-bit pointers. 32 bit is optional.

**M15.** All xSDK compatibility changes should be sustainable.

**M16.** Have a debug build option.

**M17**. Provide sufficient documentation to support use and further development.

## Recommended xSDK policies: currently encouraged, but not required

**R1.** Provide at least one validation test that can be invoked through Spack.

**R2.** Possible to run test suite under valgrind in order to test for memory corruption issues.

**R3.** Adopt and document consistent system for error conditions/exceptions.

**R4.** Free all system resources it has acquired as soon as they are no longer needed.

**R5.** Provide a mechanism to export ordered list of library dependencies.

**R6**. Provide versions of dependencies.

**R7.** Have README, SUPPORT, LICENSE, and CHANGELOG file in top directory.

**R8.** Provide version comparison preprocessor macros.

**xSDK member package**: Must be an xSDK-compatible package, *and* it uses or can be used by another package in the xSDK, and the connecting interface is regularly tested for regressions.

**We welcome feedback.
What policies make sense
for _your_ software?**

# Interoperability is challenging, particularly for deeper levels!

## Levels of package interoperability:

- **Interoperability level 1**
  - Both packages can be used (side by side) in an application
- **Interoperability level 2**
  - The libraries can exchange data (or control data) with each other
- **Interoperability level 3**
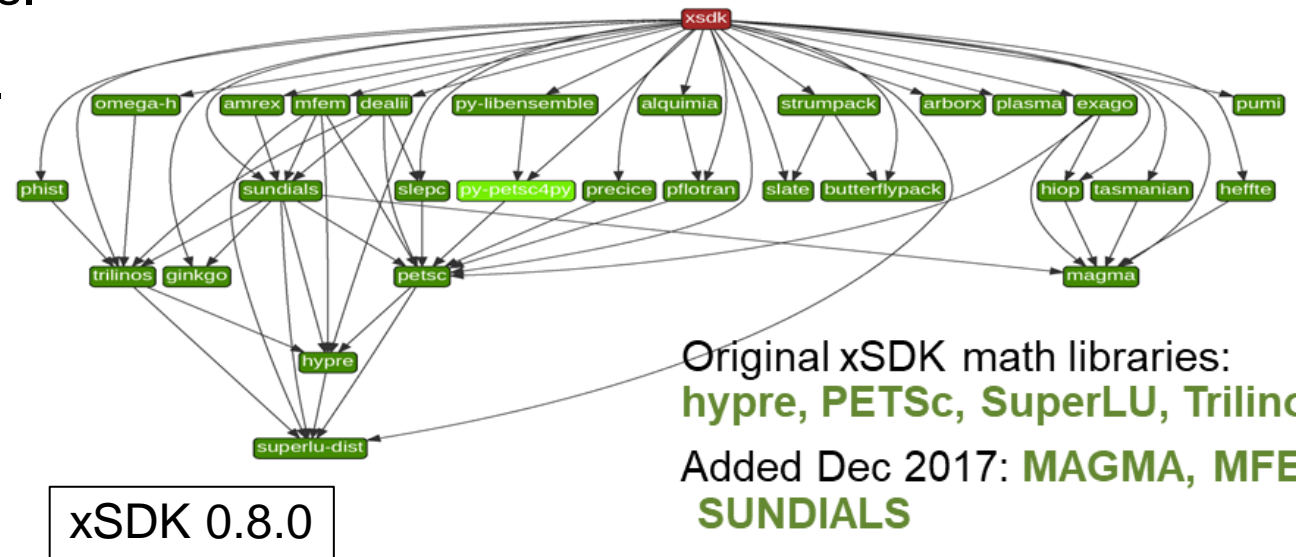  - Each library can call the other library to perform unique computations

**Notation:**

**A** ⬭→ **B**:

**A** can use **B** to provide functionality on behalf of **A**

Multiphysics Application C

Application A

Application B

SUNDIALS
$dy/dt = f(y)$

$f_A(y)$

PETSc
Linear Solver

$df_A(y)/dy$

hypre
Preconditioner

deal-ii
Finite element discretization

A

Trilinos
Linear Solver

$Ax=b$

SuperLU
Preconditioner

xSDK

ECP EXASCALE COMPUTING PROJECT
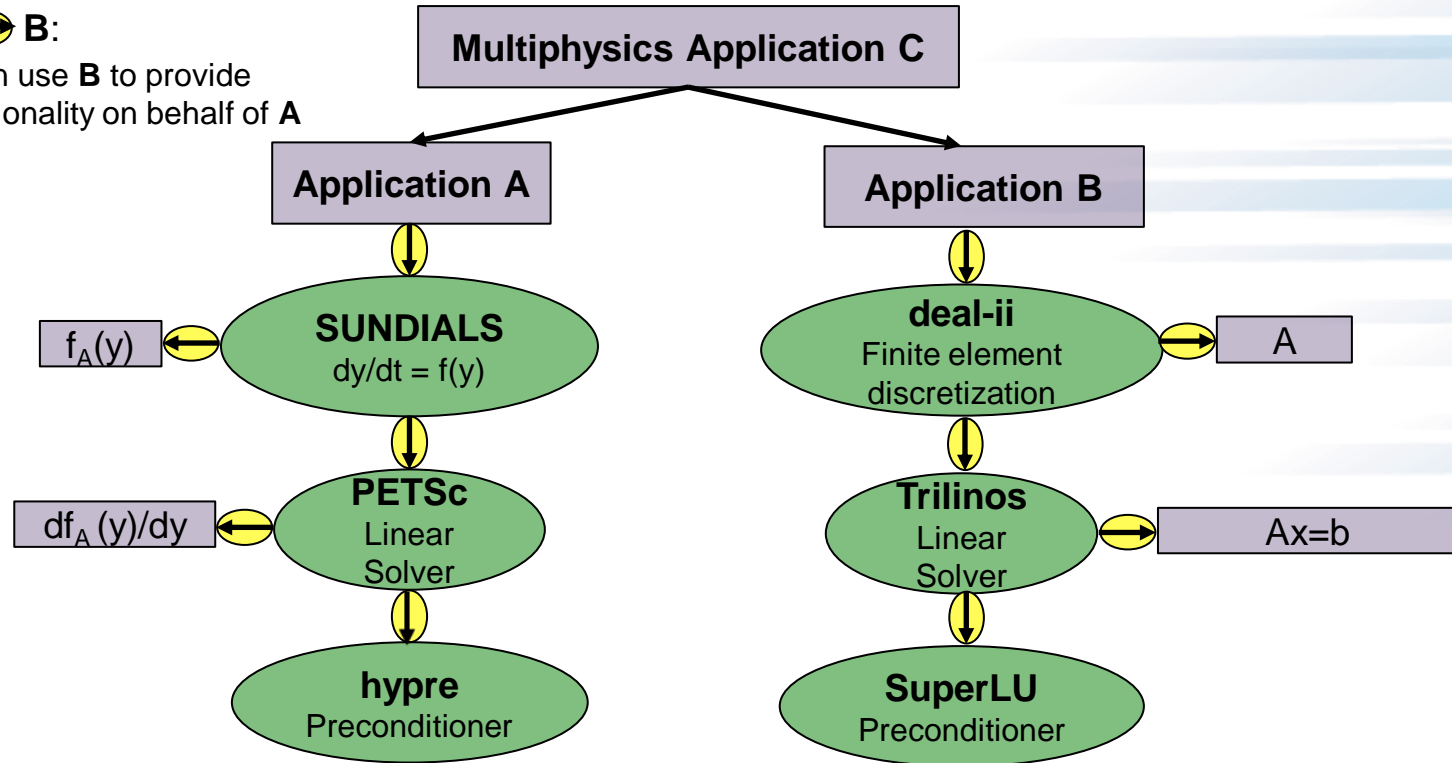
# Coordinated releases of complete xSDK with testing, documentation, packaging and deployment

- Demonstrate the impact of community policies to simplify the combined use and portability of independently developed software packages.

- Increase formality of xSDK release process.

- Expand xSDK to include additional key ECP numerical libraries as well as packages in the broader community.

- Pre-exascale environment testing:
  - Summit, Crusher (OLCF)
  - Polaris (ALCF)
  - Perlmutter (NERSC)

- Includes 9 "rocm" and 14 "cuda" enabled libraries.

- Providing specific instructions for these platforms on xSDK website
  https://xsdk.info/installing-the-software/



xSDK 0.8.0

tested on key platforms at ALCF, NERSC, and OLCF, also Linux and Mac OS X.

Original xSDK math libraries:
**hypre, PETSc, SuperLU, Trilinos**

Added Dec 2017: **MAGMA, MFEM, SUNDIALS**

Added Dec 2018: **AMReX, deal.II, DTK, Omega_h, PHIST, PLASMA, PUMI, SLEPc, STRUMPACK, TASMANIAN**

Added Nov 2019: **ButterflyPACK, Ginkgo, libEnsemble, preCICE**

Added Nov 2020: **heFFTe, SLATE**

Added Nov 2021 : **ArborX**

Added Nov 2022 : **ExaGO, HiOp**

xSDK

Spack

xSDK Member
Dependency

# Many more interoperabilities between packages exist!

|  | AMReX | ArborX | ButterflyPACK | deal-ii | DataTransferKit | ExaGO | Ginkgo | heFFTe | HiOp | hypre | libEnsemble | MAGMA | MFEM | Omega_h | PETSc | PHIST | PLASMA | preCICE | PUMI | SLATE | SLEPc | STRUMPACK | SUNDIALS | SuperLU | TASMANIAN | Trilinos |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AMReX | ■ |  |  |  |  |  |  |  |  | O |  |  |  |  | O |  |  |  |  |  |  |  | B |  |  |  |
| ArborX |  | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ButterflyPACK |  |  | ■ |  |  |  |  |  |  |  |  | O |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| deal-ii |  | O |  | ■ |  |  | O |  |  | O |  |  |  |  | B |  |  |  |  |  | B |  | O | O |  | B |
| DataTransferKit |  | O |  |  | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | B |
| ExaGO |  |  |  |  |  | ■ |  |  | O |  |  |  |  |  | O |  |  |  |  |  |  |  |  |  |  |  |
| Ginkgo |  |  |  |  |  |  | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| heFFTe |  |  |  |  |  |  |  | ■ |  |  | B | B |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| HiOp |  |  |  |  |  |  |  |  | ■ |  |  | O |  |  |  |  |  |  |  |  |  | O |  |  |  |  |
| hypre |  |  |  |  |  |  |  |  |  | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  | B |  |  |
| libEnsemble |  |  |  |  |  |  |  |  |  |  | ■ |  |  |  | B |  |  |  |  |  |  |  |  |  |  |  |
| MAGMA |  |  |  |  |  |  |  |  |  |  |  | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| MFEM |  |  |  |  |  |  | O |  | O | B |  |  | ■ | O | B |  |  |  |  |  | O | O | O | B | B |  |
| Omega_h |  |  |  |  |  |  |  |  |  |  |  |  |  | ■ |  |  |  |  |  |  |  |  |  |  |  |  |
| PETSc |  |  |  |  |  |  |  |  |  | B |  |  |  |  | ■ |  |  |  |  |  |  |  | O | O | B |  |
| PHIST |  |  |  |  |  |  |  |  |  |  |  |  |  |  | O | ■ |  |  |  |  |  |  |  |  |  |  |
| PLASMA |  |  |  |  |  |  |  |  |  |  |  | O |  |  |  |  | ■ |  |  |  |  |  |  |  |  |  |
| preCICE |  |  |  | O |  |  |  |  |  |  |  |  |  |  |  |  |  | ■ |  |  |  |  |  |  |  |  |
| PUMI |  |  |  |  |  |  |  |  |  |  |  |  |  | O |  |  |  |  | ■ |  |  |  |  |  |  | O |
| SLATE |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ■ |  |  |  |  |  |  |
| SLEPc |  |  |  |  |  |  |  |  |  |  |  |  |  |  | B |  |  |  |  |  | ■ |  |  |  |  |  |
| STRUMPACK |  |  | B |  |  |  |  |  |  |  |  | O |  |  |  |  |  |  |  | B |  | ■ |  |  |  |  |
| SUNDIALS |  |  |  |  |  |  | B |  |  | B |  |  |  |  |  |  |  |  |  |  |  |  | ■ | B |  |  |
| SuperLU |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ■ |  |  |
| TASMANIAN |  |  |  |  |  |  |  |  |  | B |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ■ |  |
| Trilinos |  |  | O |  |  |  | O | O |  | O |  | O | O |  | O |  |  |  |  | O |  | O |  | B |  | ■ |

**Legend:**

| | |
|---|---|
| O (orange) | Interoperability exists |
| B (blue) | Interoperability exists and is enabled in xSDK Spack package |

xSDK

EXASCALE COMPUTING PROJECT

# Technical Challenges



- Staying up to date while facing continual changes
  - xSDK release schedule not aligned with individual xSDK library and Spack release schedule
  - Lower dependencies can cause additional problems

- Testing difficulties
  - CI failures need to be investigated to understand what is broken and who should fix it
  - Often there is more than one package causing the issue, but finding the issues is a sequential process,
    i.e., the first issue needs to be fixed before the next one is discovered
  - The responsible package developers need to be contacted
  - Consistent oversight requires more people to respond to CI failures

- Designed test plan
  - Improve xSDK-examples test suite and integrate it with the xSDK testing process
  - Evaluate and extend current xSDK CI testing through the definition and use of hierarchical test layers, addition of new platforms and increased oversight of test results

# Multi-library example codes demonstrating interoperability

- Suite of example codes has been made available in a github repository and included in the xSDK documentation. : https://github.com/xsdk-project/xsdk-examples

- The example codes are a demonstration of interoperability between xSDK libraries and provide training for xSDK library users interested in using these capabilities.
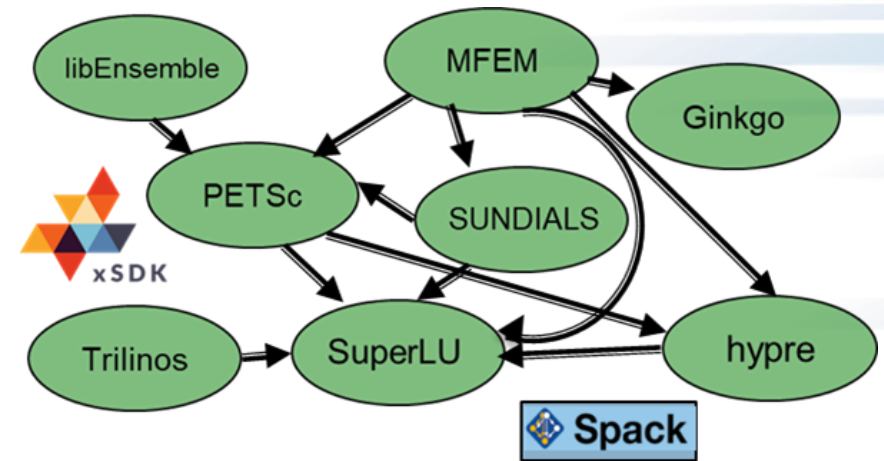
- Difficulty in building via `spack install xsdk-examples', since new interoperabilities generally not enabled in spack and/or xSDK yet. Provide simple build via `cmake'.

- Test suite important piece of xSDK testing strategy plan

# Improved xsdk-examples test suite

- The new release, xsdk-examples v0.4.0, contains the following significant improvements:
  - 24 examples codes
  - 20 xSDK libraries
  - 25 interoperabilities
  - 10 CUDA examples
  - 6 HIP examples
  - Improved build system (Spack, cmake, ctest)
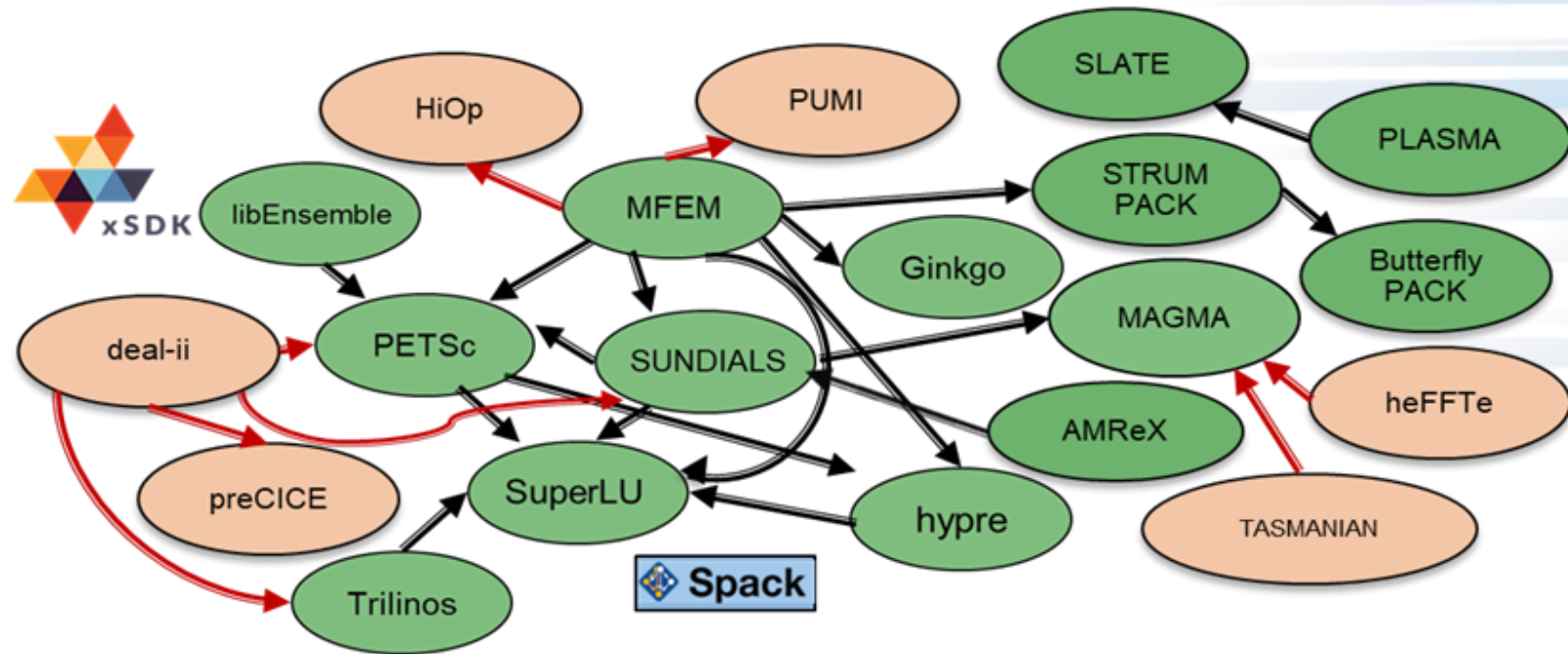
- We need to add more for future releases!!

# Hierarchical test layers

- Multi-layered testing
  - Testing strategies of the individual xSDK libraries
  - Testing of the interfaces between libraries
  - Test subsets of various interoperable packages in combination
  - Define further intermediate levels based on intricacy of library interoperability
  - Testing of the whole xSDK (final level)
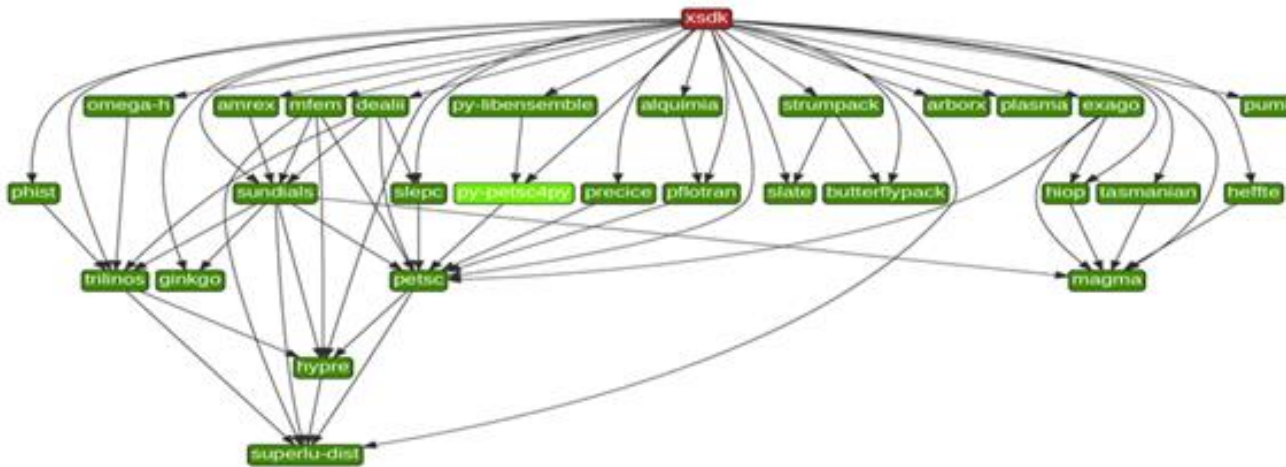
# Enhanced xSDK CI testing

- Added new platform: Linux desktop at Argonne National Lab
  - Ubuntu 20.04 with default compiler stack (gcc version 9.4.0)
  - Intel OneAPI version 2022.1.0

- Daily runs organized into 3 pipelines
  - Build xSDK and xsdk-examples on Spack development branch. This pipeline is used to identify changes in Spack that might break current xsdk, xsdk-examples releases.
    - MacOS (ANL) with gfortran/clang compilers (xsdk)
    - Linux (UTK) with GNU compilers (xsdk, xsdk-examples+cuda, xsdk-examples)
    - Linux (UTK) with Intel compilers (xsdk)
    - Linux (ANL) with GNU compilers (xsdk-examples) (new)
    - Linux (ANL) with OneAPI(Intel) compilers (new) (xsdk) (new)
  - Build xSDK with development versions of packages - on Spack development and xSDK (future) branch. This pipeline is run on the Linux server at UTK with GNU compilers. It is used to identify issues that need addressing for a future xSDK release. (This is often broken – therefore 3rd pipeline)

Pipeline    Needs    Jobs 7    Tests 0

**Test**

- ecr_l_g92-xsdk-examples-cuda-release-linux-...
- er_l_g92-xsdk-examples-release-linux-gcc920
- er_l_g94-xsdk-examples-release-linux-gcc940
- xr_l_g92-xsdk-release-linux-gcc920
- xr_l_i19-xsdk-release-linux-intel1911
- xr_l_i22-xsdk-release-linux-oneapi202210
- xr_o_c12-xsdk-release-osx-clang1200

# Testing of xSDK subsets with development versions

- Build xSDK subset of development versions of packages - on tSpack development and xSDK (future) branch, on the Linux server at UTK with GNU compilers. It contains test jobs:
  - PETSc-SLEPc-PFLOTRAN-hypre-SuperLU_dist
  - heFFTe-MAGMA-TASMANIAN

  We have added 8 new subset tests with development libraries to this pipeline
  - Libensemble-PETSc-TASMANIAN
  - MFEM-SuperLU-STRUMPACK-PETSc-Slepc-PUMI-SUNDIALS-hypre
  - MFEM-SuperLU-STRUMPACK-PETSc-Slepc-PUMI-SUNDIALS-hypre (CUDA)
  - SUNDIALS-hypre-SuperLU-PETSc
  - SUNDIALS-hypre-SuperLU-PETSc-MAGMA (CUDA)
  - AMReX-SUNDIALS
  - AMRex-SUNDIALS (CUDA)
  - Trilinos-hypre-SuperLU

Pipeline  Needs  Jobs 10  Failed Jobs 1  Tests 0

Test

- ✓ pd_l_g92-amrex-cuda-pkg-development-gcc9...
- ✓ pd_l_g92-amrex-pkg-development-gcc920
- ✓ pd_l_g92-magma-pkg-development-gcc920-h...
- ✓ pd_l_g92-mfem-cuda-pkg-development-gcc920
- ✓ pd_l_g92-mfem-pkg-development-gcc920
- ✓ pd_l_g92-petsc-pkg-development-linux-gcc92...
- ✗ pd_l_g92-sundials-cuda-pkg-development-gc...
- ✓ pd_l_g92-sundials-pkg-development-gcc920
- ✓ pd_l_g92-trilinos-pkg-development-gcc920
- ✓ pd_l_g92_libensemble-pkg-developement-gc...

# Updated Interoperability Matrix

| Color | Meaning |
|---|---|
| Orange | Interoperability exists |
| Blue | Interoperability exists and is enabled in xSDK Spack package |
| Green | Interoperability planned |
| Magenta | Interoperability exists and is enabled in xSDK Gitlab subset job |

We need to increase subsets to switch more yellow boxes to magenta ones!

Row/column packages: AMReX, ArborX, ButterflyPACK, deal-ii, DataTransferKit, ExaGO, Ginkgo, heFFTe, HiOp, hypre, libEnsemble, MAGMA, MFEM, Omega_h, PETSc, PHIST, PLASMA, preCICE, PUMI, SLATE, SLEPc, STRUMPACK, SUNDIALS, SuperLU, TASMANIAN, Trilinos

EXASCALE COMPUTING PROJECT

# Benefits of xSDK

- Improved code quality, usability, access, sustainability through compatibility with xSDK community policies

- Facilitates combined use of independently developed packages, as needed to support large-scale multiphysics and multiscale problems

- Foundation for work on deeper levels of interoperability and performance portability

# xSDK sustainability requires

- Continued collaboration between math libraries developers

- Refreshing of community policies

- Improved xSDK testing
  - xsdk-examples releases
  - Inclusion of development versions
  - Investigate and increase automation of testing

- xSDK releases with
  - New members
  - New interoperabilities between members
  - New spack options
  - Focus could be on highly connected subsets vs whole current xSDK

- All of this requires funding!

# xSDK-ECP Project Members


IT TAKES ALL KINDS TO MAKE A WORLD!

Ahmad Abdelfattah
Boyana Norris
Carol Woodward
Christian Glusa
Cody Balos
Damien Lebrun-Grandie
Daniel Arndt
Daniel Osei-Kuffuor
David Gardner
Erik Boman
Erin Carson
Gerald Ragghianti
Hartwig Anzt
Ian Mcinerney
Ichi Yamazaki
Ieva Dauzickaite
Jack Dongarra
Jamie Finney
Jennifer Loe

Jim Demmel
Jim Willenbring
Keita Teranishi
Kim Liegeois
Lois Curfman McInnes
Luc Berger-Vergiat
Mark Gates
Mike Heroux
Miroslav K. Stoyanov
Natalie Beams
Nick Higham
Osni Marques
Piotr Luszczek
Pratik Nayak
Richard Mills
Robert Falgout
Samuel Knight
Sarah Osborn
Satish Balay

Sherry Li
Siva Rajamanickam
Stan Tomov
Stephen Hudson
Stuart Slattery
Terry Cojean
Thomas Grützmacher
Tobias Ribizel
Tomas Gergelits
Tzanio Kolev
Ulrike Meier Yang
Veselin Dobrev
Victor Magri
Viktor Reshniak
Wenjun Ge
Yang Liu

And many more …

# Thank you!

This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project, a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.