

Toward Improved Sustainability of the Exascale Computing Project Software Stack



Michael A. Heroux, Sandia National Laboratories
Director of Software Technology

MAGIC Meeting, August 4, 2021

Sustainability Sketch

- A product is known to be sustainable only after it has been sustained
- But can determine attributes such as “eating broccoli” for health
- Reason to sustain (why) as important as cost/approach to sustain (how)

Sustainability \propto why \div how

ECP Software Stack:

Why sustain

- Products people use
- On emerging platforms



We work on products applications need now and into the future

Key themes:

- Focus: GPU node architectures and advanced memory & storage technologies
- Create: New high-concurrency, latency tolerant algorithms
- Develop: New portable (Nvidia, Intel, AMD GPUs) software product
- Enable: Access and use via standard APIs

Software categories:

- **Next generation established products:** Widely used HPC products (e.g., MPICH, OpenMPI, PETSc)
- **Robust emerging products:** Address key new requirements (e.g., Kokkos, RAJA, Spack)
- **New products:** Enable exploration of emerging HPC requirements (e.g., SICM, zfp, UnifyCR)

Example Products	Engagement
MPI – Backbone of HPC apps	Explore/develop MPICH and OpenMPI new features & standards
OpenMP/OpenACC –On-node parallelism	Explore/develop new features and standards
Performance Portability Libraries	Lightweight APIs for compile-time polymorphisms
LLVM/Vendor compilers	Injecting HPC features, testing/feedback to vendors
Perf Tools - PAPI, TAU, HPCToolkit	Explore/develop new features
Math Libraries: BLAS, sparse solvers, etc.	Scalable algorithms and software, critical enabling technologies
IO: HDF5, MPI-IO, ADIOS	Standard and next-gen IO, leveraging non-volatile storage
Viz/Data Analysis	ParaView-related product development, node concurrency

SLATE port to AMD and Intel platforms

ECP WBS 2.3.3.13 CLOVER (SLATE)

PI Jack Dongarra, UTK

Members UTK

Scope and objectives

- SLATE is a distributed, GPU-accelerated, dense linear algebra library, intended to replace ScaLAPACK
- SLATE covers parallel BLAS, linear system solvers, least squares, eigensolvers, and the SVD

Impact

- Initially supported NVIDIA's cuBLAS for use on current machines like Summit
- Can now use AMD's rocBLAS in preparation for Frontier, and Intel's oneMKL in preparation for Aurora
- Other projects can also leverage BLAS++ for portability

Deliverables Report: <https://www.icl.utk.edu/publications/swan-016>
Code in git repos: bitbucket.org/icl/slate/ and bitbucket.org/icl/blaspp/

Port to AMD and Intel

- SLATE and BLAS++ now support all three major GPU platforms



Accomplishment

- Refactored SLATE to use BLAS++ as portability layer
- Ported BLAS++ to AMD rocBLAS and Intel oneMKL

Key ECP Software Stack Legacy:

- Portable execution on:
 - CPUs
 - 3 different GPUs
- A bridge from CPUs to GPUs

Broader Community Engagement

*The Second Extreme-scale Scientific Software Stack Forum (E4S Forum)
September 24th, 2020, Workshop at EuroMPI/USA'20*

- Presenters from 11 institutions, 6 non-DOE
- 70 participants
 - DOE Labs, NASA
 - AMD
 - HLRS, CSCS

- E4S: The Extreme-scale Scientific Software Stack for Collaborative Open Source Software, Michael Heroux, Sandia National Laboratories
- Title: Practical Performance Portability at CSCS, **Ben Cumming, CSCS**
- Title: An Overview of High Performance Computing and Computational Fluid Dynamics at NASA, **Eric Nielsen, NASA Langley**
- Towards An Integrated and Resource-Aware Software Stack for the EU Exascale Systems, **Martin Schulz, Technische Universität München**
- Spack and E4S, Todd Gamblin, LLNL
- Rocks and Hard Places – Deploying E4S at Supercomputing Facilities, Ryan Adamson, Oak Ridge Leadership Computing Facility
- Advances in, and Opportunities for, LLVM for Exascale, Hal Finkel, Argonne National Laboratory
- Kokkos: Building an Open Source Community, Christian Trott, SNL
- Experiences in Designing, Developing, Packaging, and Deploying the MVAPICH2 Libraries in Spack, **Hari Subramoni, Ohio State University**
- Software Needs for Frontera and the NSF Leadership Class Computing Facility – the Extreme Software Stack at the Texas Advanced Computing Center, **Dan Stanzione, TACC**
- Building an effective ecosystem of math libraries for exascale, Ulrike Yang
- Towards Containerized HPC Applications at Exascale, Andrew Younge, Sandia
- E4S Overview and Demo, Sameer Shende, University of Oregon
- The Supercomputer “Fugaku” and Software, programming models and tools, **Mitsuhsa Sato, RIKEN Center for Computational Science (R-CCS), Japan**

E4S provides a natural collaboration vehicle for interacting within DOE, with other US agencies, industry and international partners

How to sustain

- Software architecture
- Quality expectations
- Access & understanding
- Plan, execute, deliver
- Better, faster, cheaper: all three



The Extreme-Scale Scientific Software Stack (E4S) and Software Development Kits (SDKs)



Extreme-scale Scientific Software Stack (E4S)



- E4S: HPC Software Ecosystem – a curated software portfolio
- A **Spack-based** distribution of software tested for interoperability and portability to multiple architectures
- Available from **source, containers, cloud, binary caches**
- Leverages and enhances SDK interoperability thrust
- Not a commercial product – an open resource for all
- Oct 2018: E4S 0.1 - 24 full, 24 partial release products
- Jan 2019: E4S 0.2 - 37 full, 10 partial release products
- Nov 2019: E4S 1.0 - 50 full, 5 partial release products
- Feb 2020: E4S 1.1 - 61 full release products
- Nov 2020: E4S 1.2 (aka, 2020.10) - 67 full release products
- Feb 2021: E4S 21.02 - 67 full release, 4 partial release
- May 2021: E4S 21.05 – 87 full release products



<https://e4s.io>

Lead: Sameer Shende
(U Oregon)



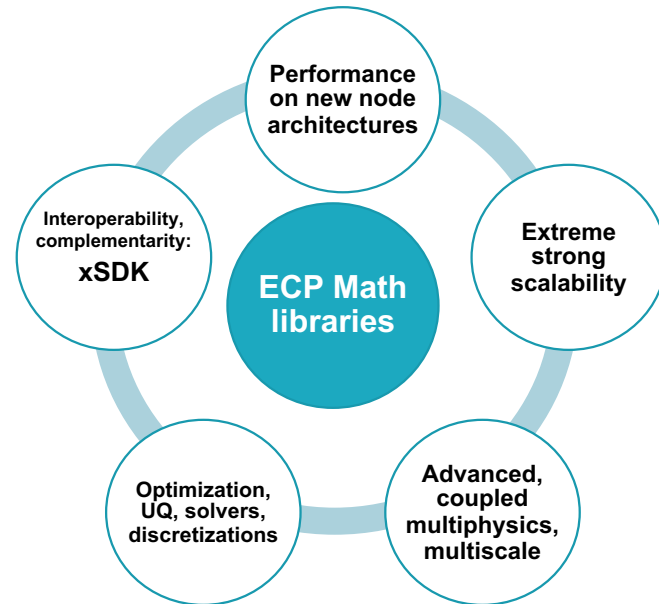
xSDK: Primary delivery mechanism for ECP math libraries' continual advancements toward predictive science

xSDK release 0.6.0 (Nov 2020)

hypr
PETSc/TAO
SuperLU
Trilinos
AMReX
ButterflyPACK
DTK
Ginkgo
heFFTe
libEnsemble
MAGMA
MFEM
Omega_h
PLASMA
PUMI
SLATE
Tasmanian
SUNDIALS
Strumpack
Alquimia
PFLOTRAN
deal.II
preCICE
PHIST
SLEPc

} from the
broader
community

As motivated and validated by
the needs of ECP applications:



Timeline:

xSDK release 1

xSDK release 2

...

xSDK release n

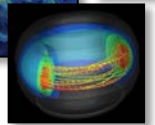
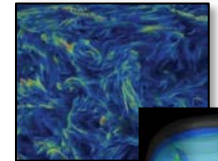
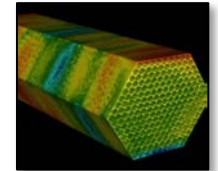
Next-generation algorithms

Advances in data structures for new node architectures

Improving library quality, sustainability, interoperability

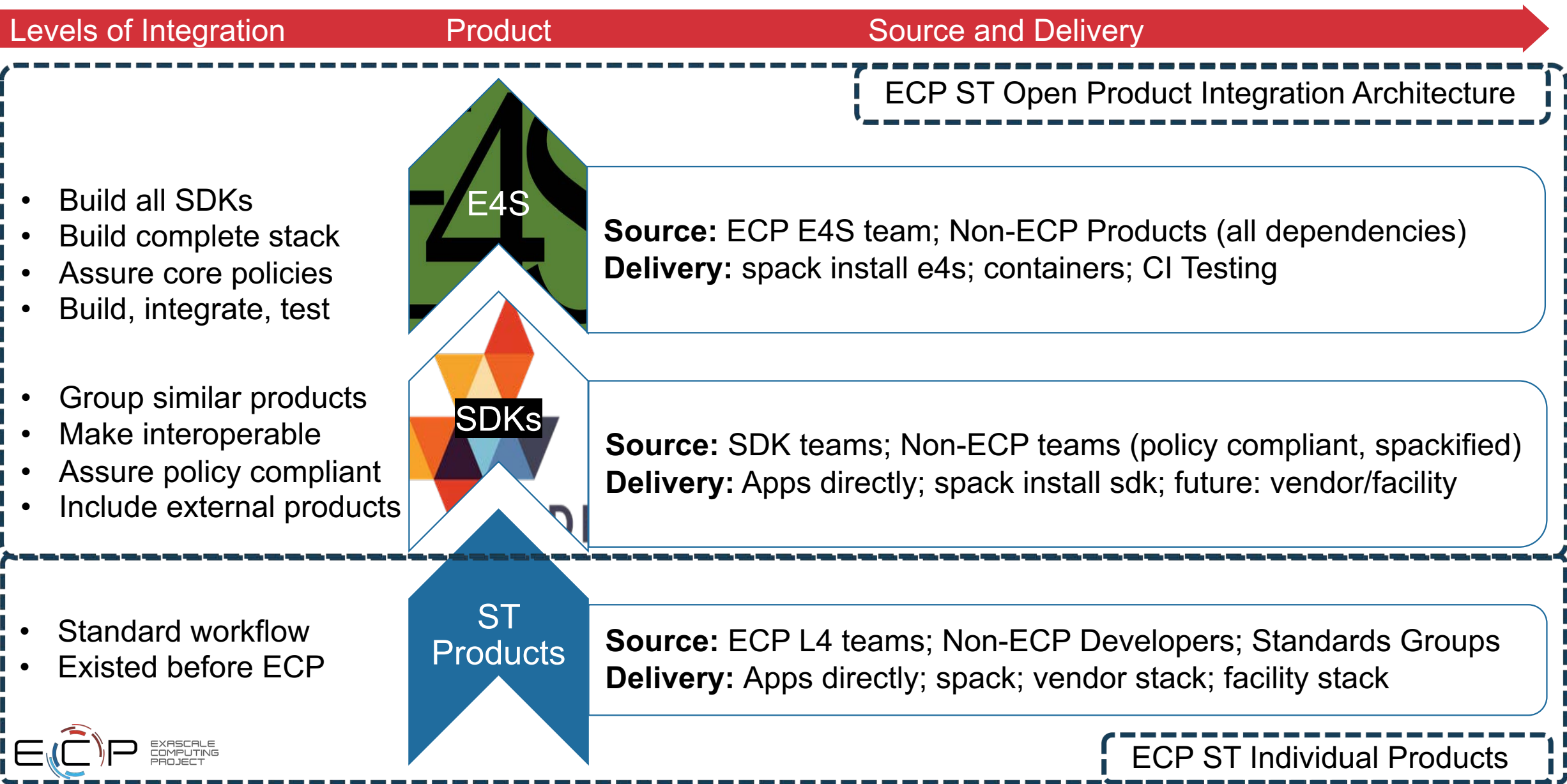
Toward predictive scientific simulations

Increasing performance, portability, productivity



Delivering an open, hierarchical software ecosystem

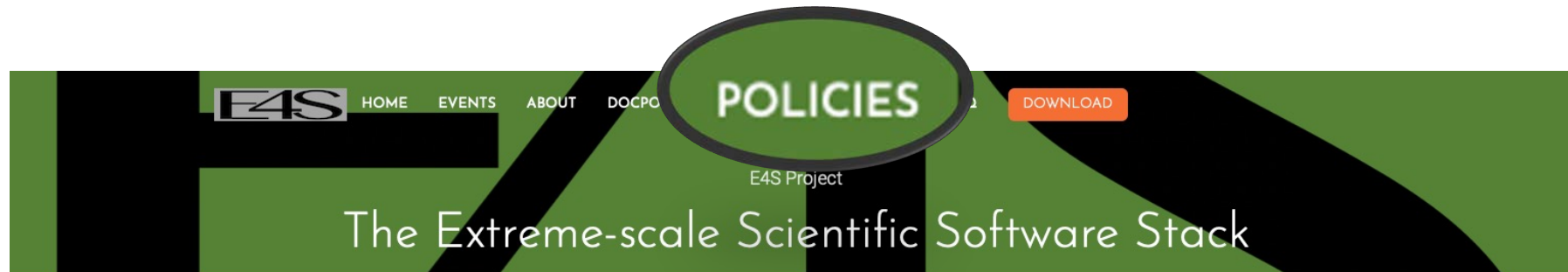
More than a collection of individual products



E4S Community Policies



E4S Community Policies V1.0 Released



What is E4S?

The Extreme-scale Scientific Software Stack (E4S) is a community effort to provide open source software packages for developing, deploying and running scientific applications on high-performance computing (HPC) platforms. E4S provides from-source builds and containers of a **broad collection of HPC software packages**.



Purpose

E4S exists to accelerate the development, deployment and use of HPC software, lowering the barriers for HPC users. E4S provides containers and turn-key, from-source builds of more than 80 popular HPC products in programming models, such as MPI; development tools such as HPCToolkit, TAU and PAPI; math libraries such as PETSc and Trilinos; and Data and Viz tools such as HDF5 and Paraview.



Approach

By using Spack as the meta-build tool and providing containers of pre-built binaries for Docker, Singularity, Shifter and CharlieCloud, E4S enables the flexible use and testing of a **large collection of reusable HPC software packages**.

E4S Community Policies Version 1

A Commitment to Quality Improvement

- Will serve as membership criteria for E4S
 - Membership is not required for *inclusion* in E4S
 - Also includes forward-looking draft policies
- Purpose: enhance sustainability and interoperability
- Topics cover building, testing, documentation, accessibility, error handling and more
- Multi-year effort led by SDK team
 - Included representation from across ST
 - Multiple rounds of feedback incorporated from ST leadership and membership
- Modeled after xSDK Community Policies
- <https://e4s-project.github.io/policies.html>

P1 Spack-based Build and Installation Each E4S member package supports a scriptable *Spack* build and production-quality installation in a way that is compatible with other E4S member packages in the same environment. When E4S build, test, or installation issues arise, there is an expectation that teams will collaboratively resolve those issues.

P2 Minimal Validation Testing Each E4S member package has at least one test that is executable through the E4S validation test suite (<https://github.com/E4S-Project/testsuite>). This will be a post-installation test that validates the usability of the package. The E4S validation test suite provides basic confidence that a user can compile, install and run every E4S member package. The E4S team can actively participate in the addition of new packages to the suite upon request.

P3 Sustainability All E4S compatibility changes will be sustainable in that the changes go into the regular development and release versions of the package and should not be in a private release/branch that is provided only for E4S releases.

P4 Documentation Each E4S member package should have sufficient documentation to support installation and use.

P5 Product Metadata Each E4S member package team will provide key product information via metadata that is organized in the *E4S DocPortal* format. Depending on the filenames where the metadata is located, this may require *minimal setup*.

P6 Public Repository Each E4S member package will have a public repository, for example at GitHub or Bitbucket, where the development version of the package is available and pull requests can be submitted.

P7 Imported Software If an E4S member package imports software that is externally developed and maintained, then it must allow installing, building, and linking against a functionally equivalent outside copy of that software. Acceptable ways to accomplish this include (1) forsaking the internal copied version and using an externally-provided implementation or (2) changing the file names and namespaces of all global symbols to allow the internal copy and the external copy to coexist in the same downstream libraries and programs. This pertains primarily to third party support libraries and does not apply to key components of the package that may be independent packages but are also integral components to the package itself.

P8 Error Handling Each E4S member package will adopt and document a consistent system for signifying error conditions as appropriate for the language and application. For e.g., returning an error condition or throwing an exception. In the case of a command line tool, it should return a sensible exit status on success/failure, so the package can be safely run from within a script.

P9 Test Suite Each E4S member package will provide a test suite that does not require special system privileges or the purchase of commercial software. This test suite should grow in its comprehensiveness over time. That is, new and modified features should be included in the suite.

E4S DocPortal



E4S DocPortal

- Single point of access
- All E4S products
- Summary Info
 - Name
 - Functional Area
 - Description
 - License
- Searchable
- Sortable
- Rendered daily from repos

E4S Products

*: Member Product

Show entries

Search:

	Name	Area	Description	
✓	ADIOS2	Data & Viz	I/O and data management library for storage I/O, in-memory code coupling and online data analysis and visualization workflows.	2021-03-10 16:45:25
✓	AML	PMR	Hierarchical memory management library from Argo.	2019-04-25 13:03:01
✓	AMREX	PMR	A framework designed for building massively parallel block- structured adaptive mesh refinement applications.	2021-05-02 17:26:43
✓	ARBORX	Math libraries	Performance-portable geometric search library	2021-01-05 15:39:55
✓	ARCHER			
✓	ASCENT			
✓	BEE	Software Ecosystem	Container-based solution for portable build and execution across HPC systems and cloud resources	2018-08-22 22:26:19
✓	BOLT	Development Tools	OpenMP over lightweight threads.	2020-05-04 11:24:57
✓	CALIPER	Development tools	Performance analysis library.	2020-11-04 23:53:07
✓	CHAI	PMR	A library that handles automatic data migration to different memory spaces behind an array-style interface.	2020-11-02 19:58:24

All we need from the software team is a repo URL + up-to-date meta-data files

Name <https://e4s-project.github.io/DocPortal.html> Latest Doc Update

Showing 1 to 10 of 76 entries Previous 1 2 3 4 5 ... 8 Next

Goal: All E4S product documentation accessible from single portal on E4S.io (working mock webpage below)

The image displays a mock-up of a web portal for E4S products. It consists of three overlapping browser window screenshots. The top-left window shows the 'e4s-project.github.io' homepage with a navigation bar containing links like 'HOME', 'EVENTS', 'ABOUT', 'DOCPORTAL', 'CONTACT US', 'FAQ', and a 'DOWNLOAD' button. The bottom-left window shows a 'Member Product' list with columns for 'Name' and 'Area', listing products like ADIOS2, AML, ARCHER, ASCENT, BEE, BOLT, CALIPER, CHAI, CINEMA, and DARSHAN. The right window shows a detailed view of the ADIOS2 product page, including its description, document summaries, and a list of ORNL researchers.

Member Product
Show 10 entries

Name	Area
ADIOS2	Data & Viz
AML	PMR
ARCHER	Tools
ASCENT	Data & Viz
BEE	Software ecosystem
BOLT	Development tools
CALIPER	Development tools
CHAI	PMR
CINEMA	Data & Viz
DARSHAN	Data & Viz

Showing 1 to 10 of 75 entries

E4S Products

* Member Product
Show 10 entries

Name	Area	Description
ADIOS2	Data & Viz	I/O and data management library for storage I/O, in-memory code coupling and online data analysis and visualization workflows.

Description: The Adaptable Input Output System version 2, developed in the Exascale Computing Program.
Homepage: <https://cmmd.ornl.gov/software/adios2>

Document Summaries

ReadMe.md

```
License: Apache 2.0
docs: [link]

Release: V2.6.0
docs: [link]
```

LICENSE

```
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
```

Computer Science and Mathematics

ADIOS2

ADIOS 2: The Adaptable Input Output (I/O) System version 2 is an open-source framework that addresses scientific data management challenges, e.g. scalable parallel I/O, as we approach the exascale era in high-performance computing (HPC). ADIOS 2 bindings are available in C++, C, Fortran, Python and can be used on supercomputers, personal computers, and cloud systems running on Linux, macOS and Windows. ADIOS 2 has out-of-the-box support for MPI and serial environments.

ADIOS 2 unified application programming interface (API) focuses on what scientific applications produce and consume in terms of n-dimensional Variables, Attributes, and Steps, while hiding the low-level details of how the data byte streams are transported as efficiently as possible from application memory to HPC networks, files, wide-area-networks, and direct memory access media. Typical use cases include file storage for checkpoint-restart and analysis, data streaming for code-coupling, and in situ analysis and visualization workflows. ADIOS 2 also provides high-level APIs that resemble native I/O libraries in Python (file) and C++ (fstream) for easy integration with their rich data analysis ecosystems. In addition, XML and YAML runtime configuration files are provided so users can fine tune available parameters to enable efficient data movements without recompiling their codes. ADIOS 2 also supports data compression via third party libraries for lossy: zfp, SZ, MGARD, and lossless: blosc, bzip2, png operations.

The ADIOS 2 development process adopts modern software engineering practices such as unit testing, continuous integration, and documentation to make the final product accessible to the scientific community. Our commitment is to release a new version every 6 months. Distributions are currently available via modern package management systems: conda, spack, homebrew (and more to come). Overall, applications using ADIOS 2 do not need to dramatically modify their source code to evaluate I/O performance trade-offs, thus reducing integration and maintenance costs in their development process. For those coming from ADIOS 1.x, ADIOS 2

ORNL Researchers

- Nicholas Thompson
- Norbert Podhorszki
- William Godoy
- Scott Klasky
- Lipeng Wan
- Jeremy Logan

Other Researchers

- Kesheng (John) Wu
Lawrence Berkeley National Laboratory
0000-0002-4907-3393
- Greg Eisenhauer
Georgia Institute of Technology
0000-0002-2070-043X
- Manish Parashar
Rutgers University
0000-0003-0963-7406
- Philip Davis
Rutgers University
0000-0002-2205-8268

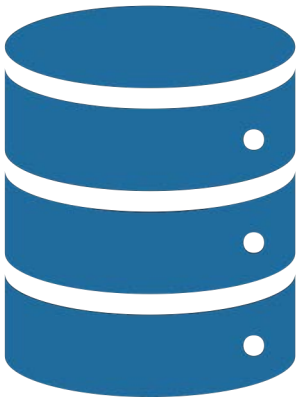
Group
Scientific Data Group

E4S Planning, Executing, Delivering



ECP ST Planning Process: Hierarchical, three-phase, cyclical

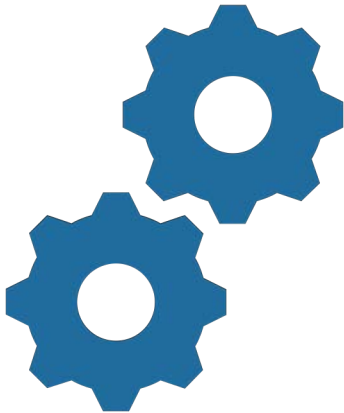
Baseline



FY20–23 Baseline Plan High level Definitions

- Q2 FY19 start
- FY20 Base plan
- FY21–23 planning packages

Annual Refinement



FY Refine Baseline Plan As Needed Basic activity definitions

- 6 months prior to FY
- 4–6 P6 Activities/year
- Each activity:
 - % annual budget
 - Baseline start/end
 - High level description

Per Activity



Detailed Plan Complete activity definitions

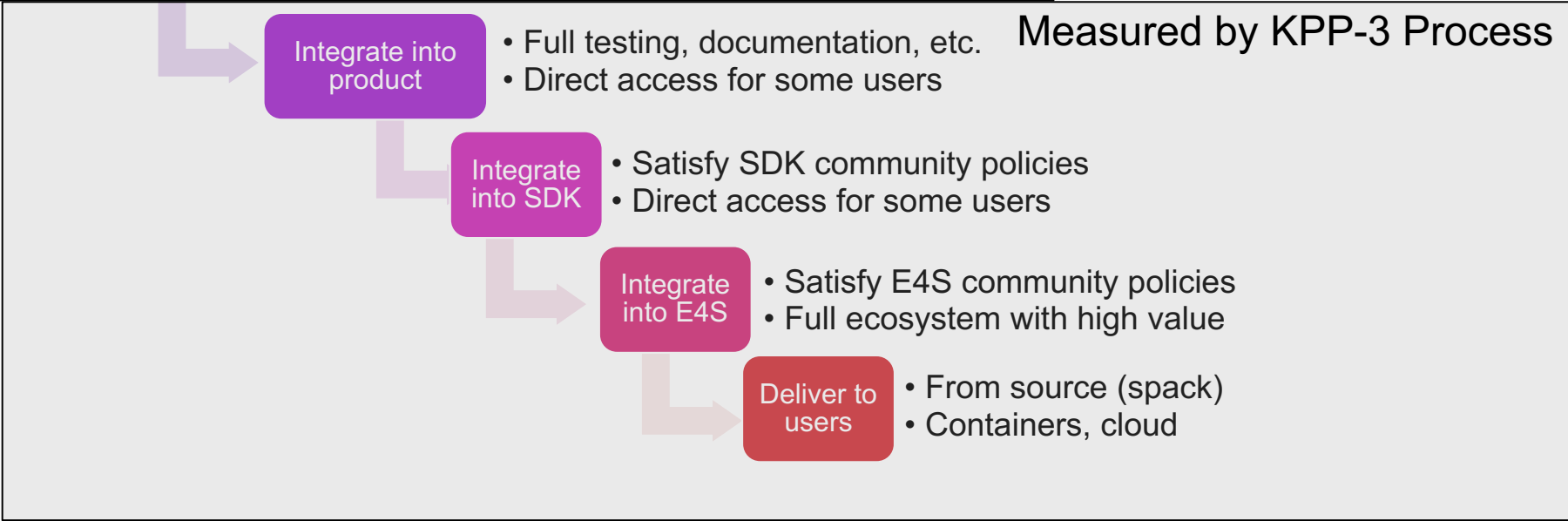
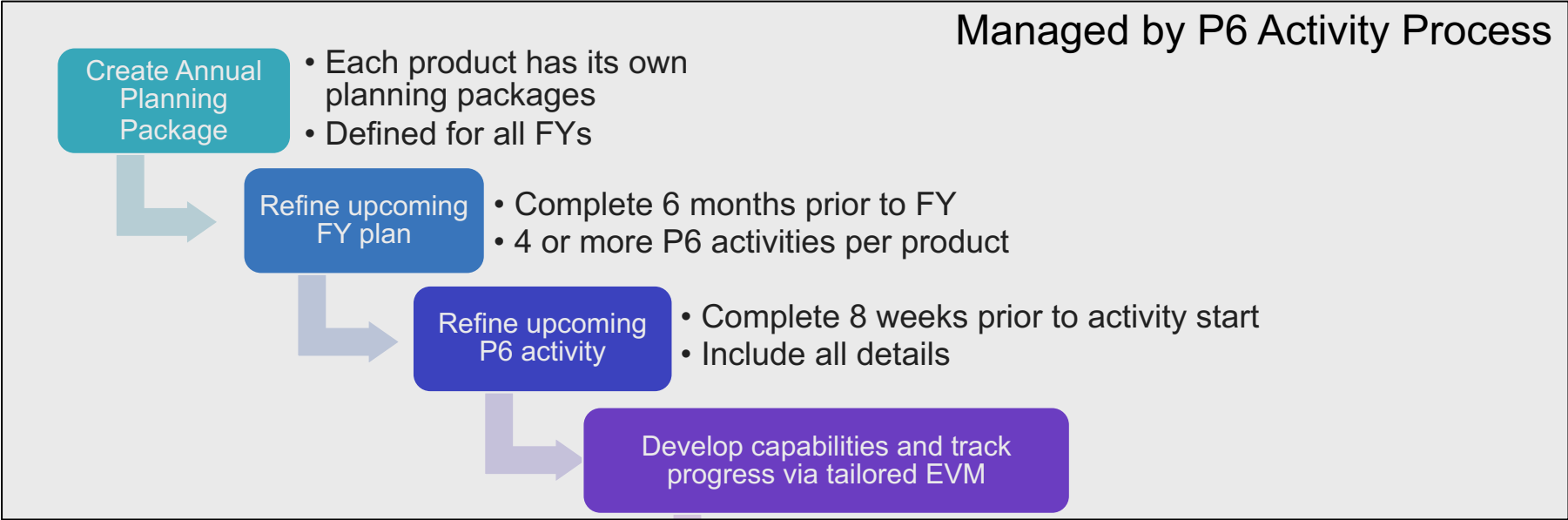
- 8 weeks prior to start
- High-fidelity description
- Execution strategy
- Completion criteria
- Personnel details

Two-level Review Process	
Changes to Cost, Scope, and Schedule	
Minor	Major
Lightweight Review in Jira, L3 and L2 leads	Change Control Board Review, ECP leadership
Variance Recorded in Jira	
Proceed with Execution	

KPP-3: Focus on capability integration

- **Capability:** Any significant product functionality, including existing features adapted to the pre-exascale and exascale environments, that can be integrated into a client environment.
- **Capability Integration:** Complete, sustainable integration of a significant product capability into a client environment in a pre-exascale environment (tentative score) and in an exascale environment (confirmed score).

ECP ST Lifecycle summary



Using E4S



Spack

- E4S uses the Spack package manager for software delivery
- Spack provides the ability to specify versions of software packages that are and are not interoperable.
- Spack is a build layer for not only E4S software, but also a large collection of software tools and libraries outside of ECP ST.
- Spack supports achieving and maintaining interoperability between ST software packages.

E4S: Spack Build Cache

E4S Build Cache for Spack 0.16.0

To use this build cache, just add it to your Spack

```
spack mirror add E4S https://cache.e4s.io  
wget https://oaciss.uoregon.edu/e4s/e4s.pub  
spack gpg trust e4s.pub
```

Click on one of the packages below to see a list of all available variants.

☒ All Architectures
 ☐ PPC64LE
 ☐ X86_64

☒ All Operating Systems
 ☐ Centos 7
 ☐ Centos 8
 ☐ RHEL 7
 ☐ RHEL 8
 ☐ Ubuntu 18.04
 ☐ Ubuntu 20.04

Last updated: 02-18-2021 14:28 PST

32658 Spack packages

Search

adiak@0.1.1
 adios2@2.5.0
 adios2@2.6.0
 adios2@2.7.0
 adios2@2.7.1
 adios@1.13.1
 adlbx@0.9.2
 adol-c@2.7.2
 amg@1.2
 aml@0.1.0
 amrex@20.07

amrex@20.09
 amrex@20.10
 amrex@20.11
 amrex@20.12
 amrex@21.01

amrex@21.02

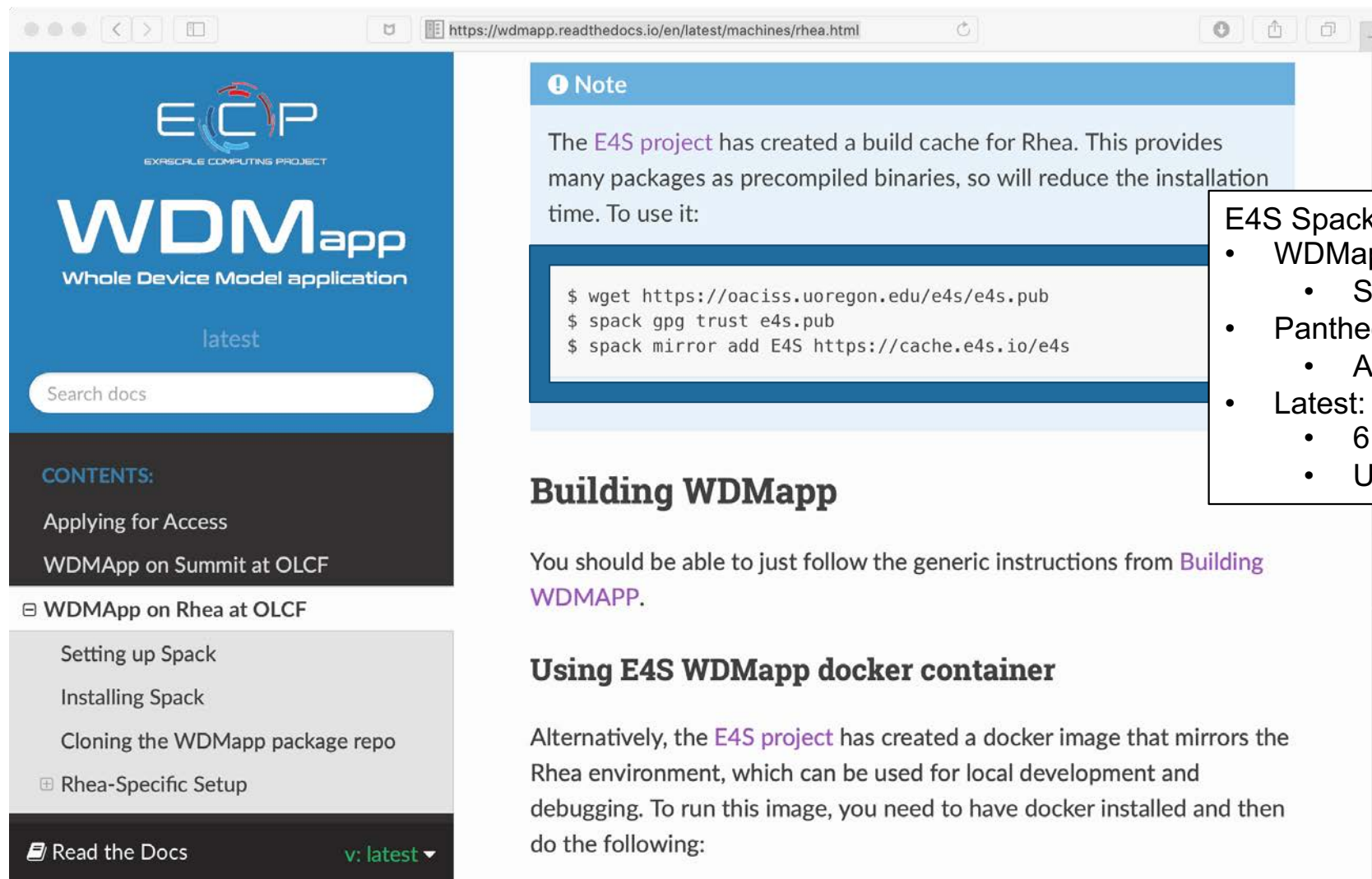
Click on the full spec link to find out more.

Link	Arch	OS	Compiler	Created	Full Hash
Full Spec	ppc64le	rhel8	gcc@8.3.1	02-03-2021 08:24 PST	ncsadwuelksqodkk2tx2aqxaupti5cz3
Full Spec	ppc64le	rhel8	gcc@8.3.1	02-08-2021 09:43 PST	y3afvxo7zbiar4xwjhj7utpwu22bzi5
Full Spec	ppc64le	ubuntu18.04	gcc@7.5.0	02-03-2021 08:23 PST	rbyn4xf344vvttl5hb7ha2saipzyod5n
Full Spec	ppc64le	ubuntu18.04	gcc@7.5.0	02-08-2021 07:44 PST	vjrj7mumpa2dbmjdlfpnxfd3fss2tjj
Full Spec	ppc64le	ubuntu20.04	gcc@9.3.0	02-03-2021 08:24 PST	2kpij4eyoxgw4g6kroxgkr2drbz6ckm
Full Spec	ppc64le	ubuntu20.04	gcc@9.3.0	02-08-2021 09:45 PST	necbis3dr4f4snsgoy3jxoelw5swvwpg
Full Spec	x86_64	ubuntu18.04	gcc@7.5.0	02-03-2021 08:03 PST	mtkbuiioif6nbawqsqg2s7iim3eccx4inleq
Full Spec	x86_64	ubuntu18.04	gcc@7.5.0	02-08-2021 07:19 PST	k6berz2tvd3l4s6rrcajgx6raszxldx
Full Spec	x86_64	ubuntu20.04	gcc@9.3.0	02-03-2021 08:04 PST	a54feyvsttm6xj2ypicuq5t7bmizp2a
Full Spec	x86_64	ubuntu20.04	gcc@9.3.0	02-08-2021 07:19 PST	dy3m43fzgoju6m42qwgg3pidlmxfj2h

ant@1.10.0
 ant@1.10.7
 arborx@0.9-beta
 argobots@1.0
 argobots@1.0rc1
 argobots@1.0rc2
 arpack-ng@3.7.0
 arpack-ng@3.8.0
 ascent@0.6.0

- 32,000+ binaries
- S3 mirror
- No need to build from source code!
- Speeds up installations 10x

WDMApp: Speeding up bare-metal installs using E4S build cache



The screenshot shows a web browser displaying the WDMApp documentation page for Rhea at OLCF. The page has a blue header with the ECP logo and 'WDMapp Whole Device Model application'. A sidebar on the left contains a 'CONTENTS' section with links like 'Applying for Access', 'WDMApp on Summit at OLCF', and 'WDMApp on Rhea at OLCF'. The main content area features a 'Note' box with instructions on using the E4S project's build cache, a code block with terminal commands, and sections for 'Building WDMapp' and 'Using E4S WDMapp docker container'.

Note

The **E4S project** has created a build cache for Rhea. This provides many packages as precompiled binaries, so will reduce the installation time. To use it:

```
$ wget https://oaciss.uoregon.edu/e4s/e4s.pub
$ spack gpg trust e4s.pub
$ spack mirror add E4S https://cache.e4s.io/e4s
```

Building WDMapp

You should be able to just follow the generic instructions from **Building WDMAPP**.

Using E4S WDMapp docker container

Alternatively, the **E4S project** has created a docker image that mirrors the Rhea environment, which can be used for local development and debugging. To run this image, you need to have docker installed and then do the following:

E4S Spack build cache:

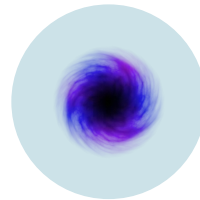
- WDMapp added E4S mirror
 - Speedup: 10X
- Pantheon: 10X
 - Another 10X via “smoother” installs
- Latest: ExaWind (Nalu-Wind)
 - 6 minutes with build cache
 - Up to 4 hours without

E4S: Better quality, documentation, testing, integration, delivery, building & use

Delivering HPC software to facilities, vendors, agencies, industry, international partners in a brand-new way



Community Policies
Commitment to software quality



DocPortal
Single portal to all E4S product info



Portfolio testing
Especially leadership platforms



Curated collection
The end of dependency hell



Quarterly releases
Release 1.2 – November



Build caches
10X build time improvement



Turnkey stack
A new user experience



<https://e4s.io>



E4S Strategy Group
US agencies, industry, international

Summary

What E4S is not

- A closed system taking contributions only from DOE software development teams.
- A monolithic, take-it-or-leave-it software behemoth.
- A commercial product.
- A simple packaging of existing software.

What E4S is

- Extensible, open architecture software ecosystem accepting contributions from US and international teams.
- Framework for collaborative open-source product integration for ECP & beyond, including AI and Quantum.
- Full collection if compatible software capabilities **and**
- Manifest of a la carte selectable software capabilities.
- Vehicle for delivering high-quality reusable software products in collaboration with others.
- New entity in the HPC ecosystem enabling first-of-a-kind relationships with Facilities, vendors, other DOE program offices, other agencies, industry & international partners.
- Hierarchical software framework to enhance (via SDKs) software interoperability and quality expectations.
- Conduit for future leading edge HPC software targeting scalable computing platforms.

Backup Content



Lessons learned from E4S/ECP ST to carry forward

- Deliver DOE reusable software as a portfolio
 - E4S value is already more than the sum of its parts
 - Community policies drive quality, membership
 - DocPortal, testing, containerization, cloud, build caches, modules, etc., greatly improve access & usability
 - Poor performing products are ID'ed, then improved or removed
- E4S is ready to extend to next-generation software and hardware needs
 - AI/ML products already in portfolio, ready for any new products
 - Quantum, FPGA, neuromorphic devices likely to be accelerators
 - From a macro software architecture, similar to GPUs
 - Software for these devices can and should be part of the same stack for holistic HPC environment
- DOE software as a portfolio is a first-class entity in the ecosystem
 - E4S planning, executing, tracking, assessing is peer collaboration with Facilities, program offices, vendors, etc
 - E4S can become a perennial asset for DOE/ASCR as part of its mission impact within and beyond DOE

E4S sustainability

Challenges

- ECP has a robust tailored 413.3b project management infrastructure
- Transitioning & adapting this infrastructure is essential for post-ECP success
- Funding models, portfolio management, org structure are particularly critical

Opportunities

- A sustainable software ecosystem for HPC software from DOE & broader community
- Payoff if done right: better, faster and cheaper – get all three

E4S Expansion – Base Scope & Gaps



Within base scope

Making a high-quality HPC product portfolio through tools, processes, and transparency

Community policies: Improve product quality upstream, shepherd membership growth

DocPortal: Provide easy access to product documentation

Portfolio testing: Protecte against regressions, prepare for new platforms

Curated collection: Maintain version compatibility across products

Turnkey stack via quarterly releases: Provide functionality via Spack, containers, clouds



Gaps not in base

Features that are a significant departure from core mission needs

Sustained support of new customers (without specific collaborative funding)

Activities related to commercial software enterprise

Ongoing support of a maintenance-only product (no longer funded for R&D)

Need: Business models for the gaps

Final points

- E4S is a curated software stack with quality improvement incentives, moving toward turnkey use
- With DOE program managers ECP is starting
 - Software ecosystem sustainability planning
 - E4S strategic plan (will include monthly townhalls)
- We believe
 - E4S has reduced important gaps that limit usefulness of DOE software for industry
 - But some gaps remain
- Next steps:
 - Better characterize these gaps
 - Explore models to further reduce and close gaps
 - Plan and execute toward sustainability

ST Capability Assessment Report (CAR)

- Tiered discussion of ECP Software Technology structure, strategy, status and plans
- From high-level overview to details about each team's activities and next steps
- Produced about twice a year
- Includes gap analyses
- E4S scope updated for emerging needs



ECP-RPT-ST-0002-2020–Public

ECP Software Technology Capability Assessment Report–Public

Michael A. Heroux, Director ECP ST
Lois Curfman McInnes, Deputy Director ECP ST
Rajeev Thakur, Programming Models & Runtimes Lead
Jeffrey S. Vetter, Development Tools Lead
Sherry Li, Mathematical Libraries Lead
James Ahrens, Data & Visualization Lead
Todd Munson, Software Ecosystem & Delivery Lead
Kathryn Mohror, NNSA ST Lead

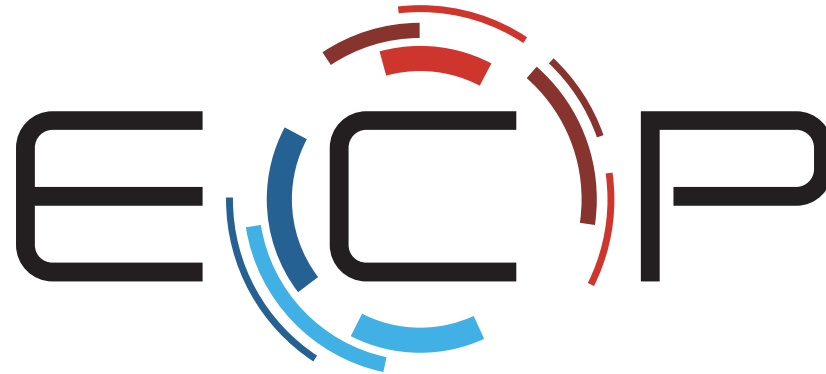
November 19, 2020

<https://www.exascaleproject.org/wp-content/uploads/2021/01/ECP-ST-CAR-v2.5.pdf>

Thank you

<https://www.exascaleproject.org>

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.



EXASCALE COMPUTING PROJECT

Thank you to all collaborators in the ECP and broader computational science communities. The work discussed in this presentation represents creative contributions of many people who are passionately working toward next-generation computational science.

"Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program."

The Networking and Information Technology Research and Development
(NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674,
Fax: 202-459-9673, Email: nco@nitrd.gov, Website: <https://www.nitrd.gov>

